



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií



REGULACE PRŮTOKU A REGULACE PRŮTOKOVÉHO OHŘÍVÁKU

Diplomová práce

Studijní program: N2612 – Elektrotechnika a informatika
Studijní obor: 3906T001 – Mechatronika

Autor práce: **Bc. Jan Šec**
Vedoucí práce: Ing. Lukáš Hubka, Ph.D.





TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies ■

THE CONTROLLER OF FLOW RATE AND THE CONTROL OF INSTANTANEOUS WATER HEATER

Diploma thesis

Study programme: N2612 – Electrical Engineering and Informatics
Study branch: 3906T001 – Mechatronics

Author: **Bc. Jan Šec**
Supervisor: Ing. Lukáš Hubka, Ph.D.



ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Jan Šec**
Osobní číslo: **M13000224**
Studijní program: **N2612 Elektrotechnika a informatika**
Studijní obor: **Mechatronika**
Název tématu: **Regulátor průtoku a regulace průtokového ohříváku**
Zadávací katedra: **Ústav mechatroniky a technické informatiky**

Z á s a d y p r o v y p r a c o v á n í :


1. Navrhněte vhodnou metodu pro číslicovou realizaci PID regulátoru.
2. Zrealizujte číslicový PID regulátor pro regulaci průtoku s možností univerzálních I/O, otestujte jej v různých režimech a ověřte jeho funkčnost na reálném zařízení.
3. Pokuste se o realizaci vzdáleného monitoringu i nastavení regulátoru prostřednictvím internetu.
4. Jako nadstavbu vytvořte jednoúčelový regulátor pro řízení systému průtokového ohříváče v konfiguraci se dvěma vstupy a dvěma výstupy a ověřte jeho funkčnost.

Rozsah grafických prací: dle potřeby dokumentace
Rozsah pracovní zprávy: cca 40–50 stran
Forma zpracování diplomové práce: tištěná/elektronická
Seznam odborné literatury:


- [1] MODRLÁK, Osvald a Lukáš HUBKA. Automatické řízení: učební text. Liberec: Technická univerzita v Liberci, 2012.
- [2] Arduino.cz - Webový magazín o Arduinu a elektronice [online]. 2014 [cit. 2014-10-07]. Dostupné z: <http://arduino.cz/>.
- [3] Arduino - Home [online]. 2014 [cit. 2014-10-07]. Dostupné z: <http://arduino.cc/>.

Vedoucí diplomové práce: Ing. Lukáš Hubka, Ph.D.
Ústav mechatroniky a technické informatiky

Datum zadání diplomové práce: 10. října 2015
Termín odevzdání diplomové práce: 16. května 2016


prof. Ing. Václav Kopecký, CSc.
děkan




doc. Ing. Milan Kolář, CSc.
vedoucí ústavu

V Liberci dne 10. října 2015

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum: 13. května 2016

Podpis:



Poděkování

Na tomto místě chci poděkovat vedoucímu mé diplomové práce Ing. Lukáši Hubkovi za ochotu, trpělivost a cenné rady při realizaci a psaní práce. Dále chci poděkovat mé rodině za poskytnutou podporu během studia a zároveň za podporu při psaní diplomové práce.

Práce byla podpořena Studentskou grantovou soutěží Technické univerzity v Liberci.



Abstrakt

V rámci diplomové práce byla řešena problematika regulace průtoku a regulace průtokového ohříváku. V úvodu je návrh a popis regulační smyčky. Následuje kapitola se stručným popisem použitých prvků, jejíž součástí je i seznámení s vybraným mikrokontrolérem – platforma Arduino Mega 2560 osazena mikroprocesorem ATmega 2560. Platforma realizuje programové řešení číslicového PID regulátoru včetně dalších operací pro zpracování a vyhodnocování dat. Následuje volba a popis realizace regulátoru průtoku. Součástí práce je i řešení problematiky spojené s regulací MIMO systému. Kapitoly obsahují blokové schéma zapojení, popis problému a následná možná řešení dané problematiky. V úplném závěru je popsána realizace monitoringu pomocí Ethernet shieldu určeného pro platformu Arduino.

Klíčová slova: Regulace, PID regulátor, mikrokontrolér, Arduino Mega 2560, Ethernet shield.

Abstract

In thesis was solve issue about flow control and regulation of the flow heater. A design and description of control loop is in introduction. Chapter with brief description of use elements follows. Familiarization with the chosen microcontroller is part of it too – platform Arduino Mega 2560 uses a microcontroller based on the ATmega 2560. This platform realizes program part of digital PID controller including operations for data processing and evaluation. Choice and description of realization of flow regulation follows. Solution of issue coupled with regulation of MIMO system is part of it too. Chapters contain block diagram, description of the problem and next are possible solutions of this issues. Description of monitoring realization with Ethernet shield which is for platform Arduino is placed in final part of this dissertation.

Key words: Regulation, PID controller, microcontroller, Arduino Mega 2560, Ethernet shield.



Obsah

1 Úvod	12
2 Obecný popis regulace průtoku	13
2.1 Návrh blokového schématu systému	13
2.2 Popis bloků použitých v blokovém schématu.....	15
2.3 Testování a nastavení výpočetních parametrů	21
2.4 Návrh softwaru pro regulaci průtoku	23
3 Regulace průtoku a teploty	32
3.1 Popis systému	32
3.2 Snímání a regulace teploty.....	33
3.3 Vliv průtoku na teplotu	38
3.4 Identifikace systému	45
3.5 Decentralizované řízení	47
4 Vzdálená komunikace.....	71
4.1 Monitoring regulace.....	72
5 Kompletace zařízení	74
6 Ovládání a obsluha zařízení.....	78
7 Závěr	80
Seznam doporučené a citované literatury	82
A Obsah přiloženého CD	84
B Arduino Mega 2560 rozložení pinů na desce [5]	85
C Provedení stromu menu	86
D Schéma zapojení celého zařízení	87

Seznam obrázků

OBRÁZEK 1: PROVEDENÍ VODNÍ SOUSTAVY V LABORATOŘI.....	13
OBRÁZEK 2: JEDNODUCHÉ BLOKOVÉ SCHÉMA REGULACE PRŮTOKU.....	14
OBRÁZEK 3: BLOKOVÉ SCHÉMA CELÉHO ZAŘÍZENÍ.....	14
OBRÁZEK 4: PONORNÉ ČERPADLO BARWING.....	16
OBRÁZEK 5: PRŮTOKOMĚR OD FIRMY BIOTECH [12]	16
OBRÁZEK 6: ELEKTRONICKÉ ZAPOJENÍ PRŮTOKOMĚRU	17
OBRÁZEK 7: POHLED NA PLATFORMU ARDUINO MEGA 2560 [3].....	17
OBRÁZEK 8: SCHÉMA ZAPOJENÍ TLAČÍTKA	19
OBRÁZEK 9: SCHÉMA ZAPOJENÍ VÝKONOVÉHO ČLENU	20
OBRÁZEK 10: PRŮBĚH VÝSTUPNÍHO SIGNÁLU Z PRŮTOKOMĚRU.....	21
OBRÁZEK 11: ČASOVÝ ÚSEK JEDNOHO PULZU	24
OBRÁZEK 12: LEVÁ OBDÉLNÍKOVÁ METODA	27
OBRÁZEK 13: LICHOBĚŽNÍKOVÁ METODA	28
OBRÁZEK 14: BLOKOVÉ SCHÉMA REGULACE TEPLoty A PRŮTOKU	32
OBRÁZEK 15: SCHÉMA ZAPOJENÍ TERMISTORU NTC	33
OBRÁZEK 16: TERMISTOR A TEPLOTNÍ ČIDLO	35
OBRÁZEK 17: SCHÉMA PŘEVODNÍKU Z PWM SIGNÁLU NA $U_{DC} = 0 - 10\text{ V}$	38
OBRÁZEK 18: NÁDOBA S VODOU, TOPNÝM TĚLESEM A TEPLOTNÍM ČIDLEM	39
OBRÁZEK 19: BLOKOVÉ SCHÉMA SISO SMYČEK S KŘÍŽOVOU VAZBOU	48
OBRÁZEK 20: SCHÉMA MODELU V PROGRAMU MATLAB & SIMULINK.....	48
OBRÁZEK 21: BLOKOVÉ SCHÉMA CELÉHO SYSTÉMU S FILTREM X_{12}	55
OBRÁZEK 22: SCHÉMA MODELU VČETNĚ FILTRU V PROGRAMU MATLAB & SIMULINK....	56
OBRÁZEK 23: PRINCIPiÁLNÍ SCHÉMA REGULÁTORU S DVĚMA STUPNI VOLNOSTI (2DOF) 65	
OBRÁZEK 24: MODEL 2DOF REGULÁTORU V PROGRAMU MATLAB & SIMULINK.....	67
OBRÁZEK 25: MODUL - ARDUINO ETHERNET SHIELD	71
OBRÁZEK 26: WEBOVÁ STRÁNKA PRO VZDÁLENÝ MONITORING.....	73
OBRÁZEK 27: PLASTOVÝ BOX.....	74
OBRÁZEK 28: POHLED DO BOXU	75
OBRÁZEK 29: STABILIZÁTOR NAPĚTÍ NA 10 V	76
OBRÁZEK 30: KRABÍČKA PRO STABILIZAČNÍ OBVOD S CHLADIČEM	77
OBRÁZEK 31: POHLED NA DISPLeJ S ÚVODNÍM MENU.....	79

Seznam grafů

GRAF 1: ZÁVISLOST PRŮTOKU NA FREKVENCÍ	23
GRAF 2: OVĚŘENÍ FUNKČNOSTI REGULACE PRŮTOKU	31
GRAF 3: PRŮBĚH TEPLoty PŘI OVĚŘOVÁNÍ FUNKČNOSTI TEPLotNÍHO ČIDLA	36
GRAF 4: ZÁVISLOST TEPLoty NA ZMĚNĚ VSTUPNÍHO NAPĚTÍ	39
GRAF 5: ZMĚNA PRŮTOKU PŘI ZMĚNĚ VSTUPNÍHO NAPĚTÍ NA ČERPADLE	40
GRAF 6: ZMĚNA VÝSTUPNÍ TEPLoty PŘI SKOKOVÉ ZMĚNĚ NAPĚTÍ NA VSTUPU TOPNÉHO TĚLESA	41
GRAF 7: ZMĚNA VÝSTUPNÍ TEPLoty MĚŘENÁ POMOCÍ TEPLotNÍHO ČIDLA A ARDUINA ..	42
GRAF 8: REGULACE PRŮTOKU POMOCÍ PI REGULÁTORU – NASTAVENÍ KONSTANTNÍHO PRŮTOKU	42
GRAF 9: ZMĚNA PRŮTOKU PŘI ZMĚNĚ VSTUPNÍHO NAPĚTÍ NA ČERPADLE	43
GRAF 10: ZMĚNA PRŮTOKU OVLIVŇUJÍCÍ TEPLotu VODY V NÁDOBĚ S TOPNÝM TĚLESEM	44
GRAF 11: PRŮBĚH TEPLoty PŘI ZMĚNĚ PRŮTOKU MĚŘENÝ TEPLotNÍM ČIDLEM (ARDUINO)	44
GRAF 12: POROVNÁNÍ SHODY NAMĚŘENÝCH HODNOT TEPLoty S NALEZENÝM PŘENOSEM F ₂₂	45
GRAF 13: POROVNÁNÍ SHODY NAMĚŘENÝCH HODNOT PRŮTOKU S NALEZENÝM PŘENOSEM F ₁₁	46
GRAF 14: POROVNÁNÍ SHODY NAMĚŘENÝCH HODNOT TEPLoty OVLIVNĚNÝCH PRŮTOKEM S NALEZENÝM PŘENOSEM F ₁₂	46
GRAF 15: REGULACE PRŮTOKU POMOCÍ PI REGULÁTORU	49
GRAF 16: REGULACE TEPLoty PID REGULÁTOREM + VLIV KŘÍŽOVÉ VAZBY	50
GRAF 17: PRŮBĚH REGULACE PRŮTOKU	51
GRAF 18: REGULACE TEPLoty ZA VYUŽITÍ MODELU SPOJENÉHO S PLATFORMOU	52
GRAF 19: PRŮBĚH REGULOVANÉ TEPLoty MĚŘENÉ POMOCÍ ARDUINA A TEPLotNÍHO ČIDLA	52
GRAF 20: REGULACE PRŮTOKU POMOCÍ ARDUINA	54
GRAF 21: REGULACE TEPLoty POMOCÍ ARDUINA	54
GRAF 22: PRŮBĚH REGULACE PRŮTOKU – MODEL	57
GRAF 23: PRŮBĚH REGULACE TEPLoty OPATŘENÝ FILTREM	57
GRAF 24: REGULACE PRŮTOKU POMOCÍ PI REGULÁTORU A ARDUINA	60
GRAF 25: REGULACE TEPLoty POMOCÍ PID MODELOVÉHO REGULÁTORU + FILTR	60
GRAF 26: PRŮBĚH TEPLoty NAMĚŘENÝ POMOCÍ ARDUINO TEPLotNÍHO ČIDLA	61
GRAF 27: REGULACE PRŮTOKU POMOCÍ ARDUINA	62
GRAF 28: REGULACE TEPLoty POMOCÍ ARDUINA S FILTREM	62
GRAF 29: POROVNÁNÍ MODELOVÝCH A REÁLNYCH PRŮBĚHŮ - REGULACE PRŮTOKU	63
GRAF 30: POROVNÁNÍ MODELOVÝCH A REÁLNYCH PRŮBĚHŮ - REGULACE TEPLoty	64
GRAF 31: PRŮBĚH AKČNÍCH ZÁSAHŮ - REGULACE PRŮTOKU	64
GRAF 32: PRŮBĚH AKČNÍCH ZÁSAHŮ - REGULACE TEPLoty	65
GRAF 33: PRŮBĚH REGULACE PRŮTOKU – MODEL	68
GRAF 34: PRŮBĚH REGULACE TEPLoty ZA POMOCI 2DOF REGULÁTORU	68
GRAF 35: REGULACE PRŮTOKU POMOCÍ PI REGULÁTORU A ARDUINA	69

GRAF 36: PRŮBĚH REGULACE TEPLoty POMOCÍ MODELU S 2DOF MODELOVÝM REGULÁTOREM.....	70
GRAF 37: PRŮBĚH REGULACE TEPLoty NAMĚŘENÝ TEPLOTNÍM ČIDLEM S ARDUINEM....	70

Seznam tabulek

TABULKA 1: NAMĚŘENÉ A VYPOČTENÉ HODNOTY PRŮTOKU	22
TABULKA 2: HODNOTY KONSTANT PRO STEINHART- HART ROVNICI.....	34
TABULKA 3: NAMĚŘENÉ HODNOTY TEPLoty A ODPORŮ PŘI FUNKČNÍCH TESTECH	36

Seznam zdrojových kódů

ZDROJOVÝ KÓD 1: STANOVENÍ DOBY JEDNOHO PULZU	24
ZDROJOVÝ KÓD 2: VÝPOČET PERIODY, FREKVENCE A PRŮTOKU	25
ZDROJOVÝ KÓD 3: VÝPOČET PROPORCIONÁLNÍHO ČLENU	26
ZDROJOVÝ KÓD 4: PROGRAMOVÉ ŘEŠENÍ INTEGRAČNÍ SLOŽKY (OBDÉLNÍKOVÁ METODA)	28
ZDROJOVÝ KÓD 5: PROGRAMOVÉ ŘEŠENÍ INTEGRAČNÍ SLOŽKY (LICHOBĚŽNÍKOVÁ METODA).....	29
ZDROJOVÝ KÓD 6: PROGRAMOVÉ ŘEŠENÍ DERIVAČNÍ SLOŽKY	29
ZDROJOVÝ KÓD 7: PROGRAMOVÉ ŘEŠENÍ DERIVAČNÍ SLOŽKY (FILTROVANÁ DERIVACE)	30
ZDROJOVÝ KÓD 8: VÝPOČET TEPLoty POMOCÍ STEINHART- HART ROVNICE.....	35
ZDROJOVÝ KÓD 9: PROGRAMOVÉ ŘEŠENÍ MAC A IP ADRES.....	72
ZDROJOVÝ KÓD 10: UKÁZKA KÓDU VYKRESLUJÍCÍHO DATA NA INTERNET	73

1 Úvod

Diplomová práce vznikla za účelem obnovit a inovovat experimentální úlohu v laboratorní učebně TK4. Práce je rozdělena na čtyři hlavní kapitoly. První kapitola se zabývá částí systému, který je určen pro regulaci průtoku kapaliny. Součástí kapitoly je schéma popisující základní princip systému, s kterým se pracuje. Následuje návrh blokového schématu určeného pro regulaci průtoku. Dále kapitola zahrnuje volbu a provedení mikrokontroléru, který zde zajišťuje realizaci číslicového regulátoru. Regulátor je řešen softwarovou podobou ve zmíněném mikrokontroléru. Softwarová část je realizována programovacím jazykem, který je obdobou jazyku C++.

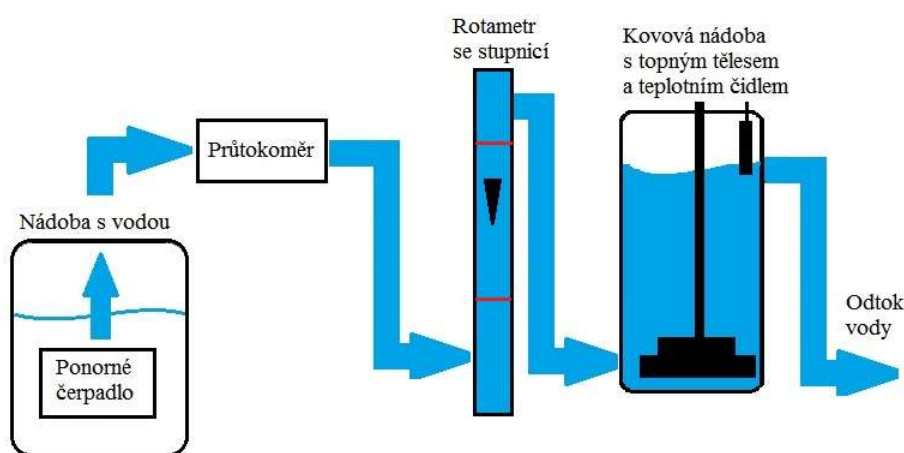
Druhá kapitola řeší úlohu v podobě systému s dvěma vstupy a dvěma výstupy. V úvodu je blokové schéma celého systému včetně popisu. Součástí systému jsou dva regulátory, které slouží k regulaci průtoku a teploty kapaliny. Řeší se problematika spojená s řízením více rozměrových systémů obsahujících takzvané křížové vazby, které s sebou nesou negativní dopad na regulaci celého systému. V kapitole je řešena identifikace systému spolu s metodami vedoucími na eliminaci křížových vazeb. Součástí kapitoly je řešení a realizace teplotního čidla umožňující měření teploty vody.

Závěrečné kapitoly pak řeší vzdálený monitoring regulace. Řešení je v podobě mikrokontrolérové nadstavby. Následující kapitola je věnována volbě boxu, do kterého se vloží zhotovené desky plošných spojů. Spolu s boxem je zde řešena i realizace napájecího zdroje k zařízení. Závěrečná kapitola slouží jako návod či manuál, který je určen pro práci se zařízením. V návodu je popis zapojení celého zařízení včetně vstupních a výstupních svorek a příslušných vodičů. Součástí návodu je i popis programového menu, volba parametrů a popis funkcí, které je možné měnit či vyvolat pomocí stisku jednotlivých tlačítek.

Cíle, kterých bych chtěl v práci dosáhnout, jsou zrealizovat regulaci průtoku za pomoci PID regulátoru. Dalším krokem je zrealizovat teplotní čidlo pro měření teploty a následně realizovat regulaci více rozměrového systému s dvěma vstupy a výstupy. Umožnit sledování průběhu regulace přes internet s případnou možností změny parametrů. Zařízení zkompletovat a odzkoušet správnost funkčnosti v laboratoři.

2 Obecný popis regulace průtoku

Celá vodní soustava začíná čerpadlem, které je ponořené do nádoby s kapalinou. Kapalina, v tomto případě se jedná o vodu, je hnána přes turbínkový průtokoměr s Hallovou sondou. Odtud je voda dále hnána do skleněného rotametrů s vyznačenou stupnicí. Stupnice zde slouží jako ukazatel minimální a maximální hodnoty průtoku. Jde o hodnoty, mezi kterými se pohybuje žádaný průtok. Z rotametrů se voda dostává do kovové nádoby ve tvaru válce. Spolu s vodou je ve válci ještě umístěno topné těleso, kterým se ohřívá přitékající voda. V horní části válce je umístěno teplotní čidlo, které měří teplotu vody, která z válce vytéká pryč a nevrací se zpět. Jedná se tedy o otevřenou vodní soustavu, kdy se vytékající voda nevrací zpět do nádoby s čerpadlem. Takto realizovaná soustava se dá rozdělit na dvě části. Obě části je možné regulovat pomocí žádaných a aktuálně získaných hodnot. Popisovaná soustava je zobrazena na obrázku 1.



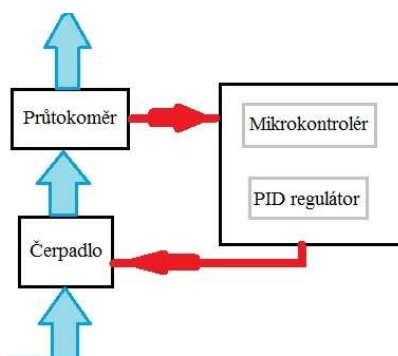
Obrázek 1: Provedení vodní soustavy v laboratoři

2.1 Návrh blokového schématu systému

Jedním z požadavků na diplomovou práci je regulace průtoku pomocí PID regulátoru. Pro samotný návrh blokového schématu se vyjde z obrázku 1. K regulaci průtoku se využijí prvky v podobě ponorného čerpadla, průtokoměru a rotametrů se stupnicí.

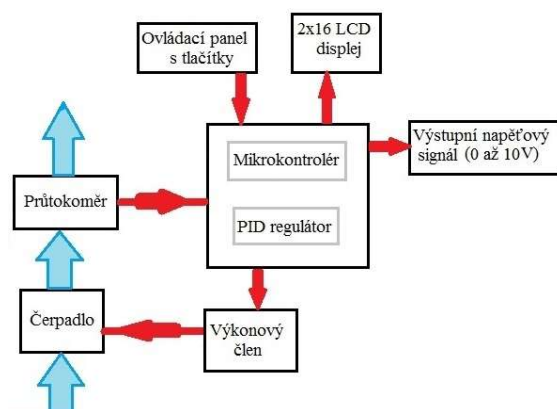
V této kapitole je tedy řešena problematika a způsob realizace regulace průtoku kapaliny. Nyní je třeba provést jednoduchý návrh blokového schématu, který je na obrázku 2. Základní stavební dvojicí je ponorné čerpadlo ženoucí kapalinu do průtokoměru. Dle

rychlosti čerpadla se průtok, měřený průtokoměrem, zmenšuje či naopak zvětšuje. Výstupní data z průtokoměru jsou v podobě digitálního signálu. Výstupní signál z průtokoměru je přiveden do mikrokontroléru, který jej zpracuje a vhodně upraví pro další vyhodnocování. Součástí mikrokontroléru je i číslicově realizovaný PID regulátor. Výstup z regulátoru, respektive výstup z mikrokontroléru, je přiveden na čerpadlo.



Obrázek 2: Jednoduché blokové schéma regulace průtoku

Dalším krokem je rozšíření blokového schématu z obrázku 2. Výsledné rozšíření je následně zobrazeno na obrázku 3, kde je vidět celá regulační smyčka se všemi potřebnými bloky, včetně popisu celého schématu. Modré šipky reprezentují tekoucí kapalinu a červené šipky reprezentují komunikaci mezi jednotlivými bloky celého schématu.



Obrázek 3: Blokové schéma celého zařízení

První dva bloky zůstaly beze změny. Základní dvojice je tedy opět tvořena ponorným čerpadlem a průtokoměrem. Ovšem blok s mikrokotrolérem se rozrostl a rozšířil o několik dalších bloků. Přibyl blok s dvouřádkovým displejem, kde je možné sledovat žádané hodnoty. Dalším rozšiřujícím blokem, jak již bylo zmíněno výše, je ovládací panel obsahující sadu čtyř tlačítek. Tlačítka jsou určena pro pohyb v menu a změnu velikosti vstupních parametrů. Přibyl blok pro výstupní napěťový signál, který je v rozmezí 0 – 10 V. Velikost tohoto signálu je závislá na velikosti aktuálního průtoku. Při nastaveném maximálním průtoku se dostane i maximální napěťový signál a obráceně. Posledním nově přidaným blokem, je blok s výkonovým členem. S pomocí výkonového členu je možno malým napětím a proudem spínat větší napětí a hlavně velké proudy, kterými čerpadlo disponuje, což je dáno samotnými parametry čerpadla, ale tím se bude zabývat následující podkapitola. Zároveň je součástí kapitoly podrobný popis jednotlivých bloků z obrázku 3 včetně jejich funkce.

2.2 Popis bloků použitých v blokovém schématu

2.2.1 Ponorné čerpadlo

První blok, který je zde popsán, je průmyslově vyráběné ponorné čerpadlo. Jedná se o poměrně malé čerpadlo s typovým označením Barwing 10 L/MIN, které je na obrázku 4. Jde sice o rozměrově malé, ale přesto výkonné čerpadlo, vhodné pro mobilní použití. Jak je z názvu patrné, maximální čerpací množství je deset litrů za minutu. Udávaná čerpací výška činí 6 m. Napájení v podobě stejnosměrného napětí se pohybuje v rozmezí od 6 V do 12 V s příkonem proudu v rozmezí 0,9 A až 1,7 A. Maximální 12V napájení je možné při provozu čerpadla pouze 30 minut. Pro trvalý provoz s podmínkou vodního chlazení je uváděno napětí od 6 V do zhruba 10 V [11].



Obrázek 4: Ponorné čerpadlo Barwing

2.2.2 Průtokoměr

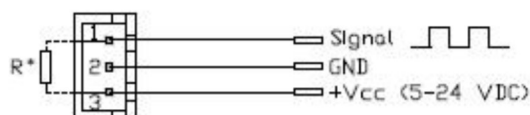
Pro měření průtoku je zde využito průmyslově vyráběného průtokoměru FCH-m-POM-LC od firmy B. I. O- TECH e. K, které lze vidět na obrázku 5. Průtokoměr je ideální pro měření malých průtoků oleje, benzínu, nafty a vody.



Obrázek 5: Průtokoměr od firmy BIOTECH [12]

Princip měření průtoku za pomoci průtokoměru je založen na malé turbíně s Hallovou sondou. Rozsah průtoku je udáván od 0,05 l/min až do 3,0 l/min při teplotě 20 °C a použitém průměru trysky 3 mm. Dále výrobce udává frekvenci 2500 impulzů na jeden litr, což je případ pro tuto situaci včetně průměru trysky. Ovšem pokud by byla použita tryska o průměru 1 mm, byla by frekvence, kterou rovněž výrobce uvádí, až 10000 impulzů na jeden litr [9]. Průtokoměr je opatřen výstupem, který tvoří celkem tři piny. Elektrické zapojení, které je na obrázku 6, není složité. Avšak mezi signálový pin jedna a napěťový pin tři je nutné zapojit rezistor s odporovou hodnotou v rozmezí 1,6 k Ω až 2,2 k Ω . Zde je využita hodnota rezistoru o velikosti 2 k Ω . Jedná o takzvaný pull up rezistor, pomocí kterého nastavujeme logickou jedničku, či logickou nulu. Díky tomu se dostává na výstupu z průtokoměru signál, který je v podobě obdélníkových pulsů. První pin

slouží pro výstupní signál jdoucí z průtokoměru a zbylé dva piny slouží pro napájení. Průtokoměr je potřeba napájet stejnosměrným napětím v rozmezí 5 V až 24 V. V tomto případě je zvoleno napájecí napětí 5 V. Zapojení pull up rezistoru včetně popisu zapojení jednotlivých pinů je na obrázku 6.



Obrázek 6: Elektronické zapojení průtokoměru

2.2.3 Mikrokontrolér

Výběr mikrokontroléru padl na platformu Arduino Mega 2560, která je na obrázku 7.



Obrázek 7: Pohled na platformu Arduino Mega 2560 [3]

Deska je osazena procesorem ATmega 2560. Platforma má 54 digitálních vstupů a výstupů. Z toho 15 může být použito jako PWM výstup. Dále je zde 16 analogových vstupů a čtyři hardwarové sériové porty. Na desce se nachází rovněž krystalový oscilátor o frekvenci 16 MHz. Celá deska je napájena pomocí napěťového konektoru jack, nebo pomocí USB kabelu. USB kabel zde slouží nejen pro napájení, ale i pro nahrávání programů do Arduina. Nedílnou součástí je i mikrospínač, který slouží k resetování. Výstupní stejnosměrné napětí, které z Arduina je možné dostat je 3,3 V a 5 V. Vstupní stejnosměrné napětí, které napájí platformu Arduino, je v rozmezí od 7 V do 12 V, pokud se nevyužívá napájecí USB kabel [3].

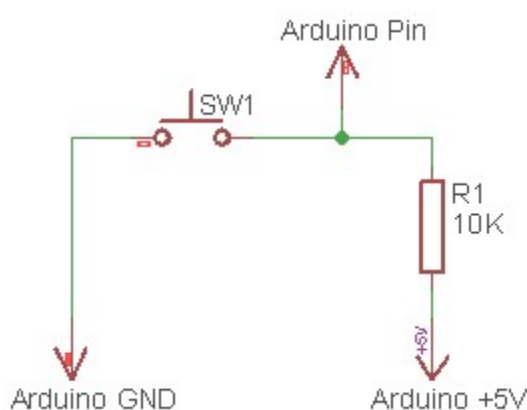
Nespornou výhodou je velký počet vstupních a výstupních pinů, kterými Arduino Mega 2560 disponuje. Další výhodou je výpočetní výkon, který je platformou poskytován. S tím úzce souvisí i poměr výkon a cena, kdy se dá platforma Arduino pořídit za relativně nízkou cenu. Dále zde odpadá nutnost programátoru, který umožňuje nahrání programu do mikrokontroléru. Zde se vystačí pouze s USB kabelem připojeným k Arduino a softwarem pro psaní programové části, pomocí něhož se celý program odladí a následně nahraje do mikroprocesoru. Další výhodou je vydávaný software, který je takzvaný open source. Tuto výhodu ocení především zkušení programátoři. Platforma je navíc velmi rozšířená ve všech možných provedeních a velikostech. S tím souvisí i široká možnost podpory ze strany výrobce a programátorů.

Prvotní volba mikrokontroléru padla na PIC16F88. Ovšem mikrokontrolér PIC nedisponuje potřebným počtem digitálních vstupů a výstupů, které jsou potřeba k finální realizaci. Další nevýhodou je pak potřeba programátoru, pomocí něhož se nahraje program do mikrokontroléru. Ceny programátorů určených pro tento typ mikroprocesorů nejsou zrovna nízké a u podomácku vyrobených programátorů pak není zaručena správná funkčnost. Zároveň je třeba navrhnout desku plošných spojů, na které je mikrokontrolér spolu s dalšími součástkami umístěn. S tím souvisí problém vznikající při osazení desky plošných spojů, jelikož některé mikrokontroléry jsou náchylné na vysokou teplotu, která vzniká během osazování. Vysoká hodnota teploty může vést k poškození mikrokontroléru. Další výhodou platformy Arduino je fakt, že je určena pro začínající, ale i pro pokročilé uživatele. Z těchto důvodů se upustilo od realizace pomocí PIC a přešlo se k volbě, která padla na platformu Arduino, což se ukázalo jako jedna ze správných možností.

2.2.4 Periferie pro mikrokontrolér

Konkrétně se jedná se o bloky, které jsou připojeny k Arduino a slouží pro zobrazení, nebo změnu daných parametrů. Jedním z nich je LCD displej od firmy Winstar s typovým označením WH1602L1-YYH-ET. Jde o dvouřádkový displej, kde ke každému z obou řádků přísluší celkem 16 znaků. Displej je opatřen vstupními a výstupními piny, pomocí nichž je napájen. Zároveň je displej přes tyto výstupní piny připojen k Arduino a tak umožňuje zobrazovat požadované hodnoty a informační texty. Nedílnou součástí displeje je standardně používaný řadič s typovým označením HD44780.

Druhým podstatným prvkem pro ovládání Arduina, zajišťující pohyb v menu, nebo změnu velikosti vstupních hodnot, je soubor čtyř tlačítek s příslušným informačním potiskem, který vypovídá o jejich funkci. První tlačítko slouží k potvrzování, jedná se o takzvaný *Enter*. Další dvě tlačítka jsou potištěna symbolem ve tvaru šipky směřující nahoru a dolů. Ty umožňují zvyšovat či snižovat velikost žádané hodnoty a dále zajišťují onen zmíněný pohyb v menu. Poslední tlačítko slouží jako návratové tlačítko. Umožňuje se vrátet zpět a je označené symbolem *Esc*. Všechny čtyři tlačítka jsou na společné desce plošných spojů a jsou propojena příslušnými piny k Arduinu.

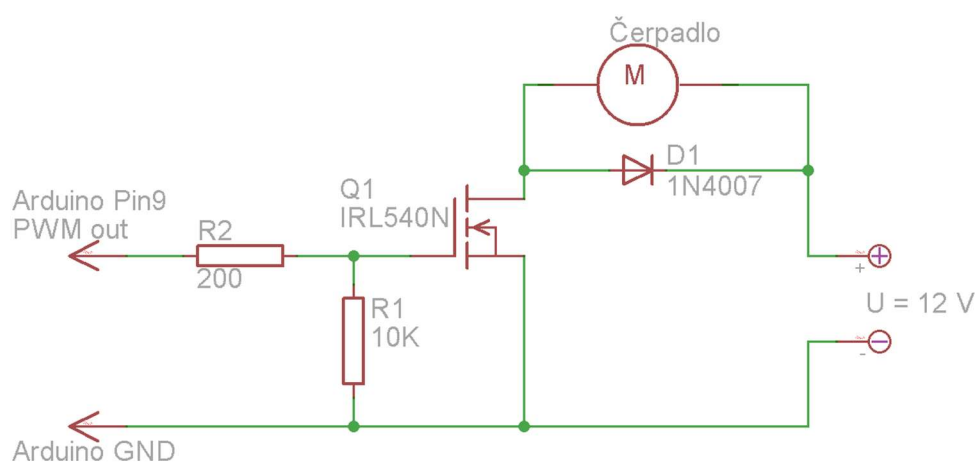


Obrázek 8: Schéma zapojení tlačítka

Na obrázku 8 je schéma, podle kterého jsou zapojena všechna výše popsaná tlačítka. Jedna strana tlačítka je připojena na *GND* pin, který je vyveden z Arduino desky a druhá strana tlačítka je přes $10\text{k}\Omega$ pull up rezistor připojena na napájecí pin s napětím 5 V. Výsledkem je přivedení logické jedničky na vstupní pin při rozepnutém tlačítku a logické nuly při sepnutém tlačítku. Elektrický bod spojující tlačítko s rezistorem je připojen k příslušnému Arduino pinu. Každé tlačítko je připojeno k jinému vstupnímu pinu a podle tohoto pinu dochází k rozpoznání stavu a typu tlačítka. Po stisknutí příslušného tlačítka dojde k vyhodnocení a k následnému vyvolání příslušné funkce, která je přiřazena danému tlačítku.

2.2.5 Výkonová část

Posledním blokem je blok s výkonovou částí. Vstupem do bloku jsou malé hodnoty napětí a proudů a výstupem jsou vyšší napěťové hodnoty, ale hlavně několikanásobně větší hodnoty proudů, což je způsobeno připojeným ponorným čerpadlem, jelikož hodnoty proudu se pohybují okolo 1 A. Spínání výkonu, se řeší pomocí jednoduchého schématu, jak je vidět z obrázku 9. Pro spínání velkých výkonů se využívá MOSFET tranzistor. Jedná se o unipolární tranzistor, který je řízený polem. Mezi přednosti patří velmi vysoký vstupní odpor, malý řídicí příkon a velký rozsah kolektorových proudů. Navíc disponuje dobrým odvodem ztrátového výkonu tranzistoru z čipu. Řízení tranzistoru probíhá pomocí napětí přiváděného na řídicí elektrodu. Ochrana proti proražení je v podobě ochranné diody, která je integrována přímo na čipu [6].



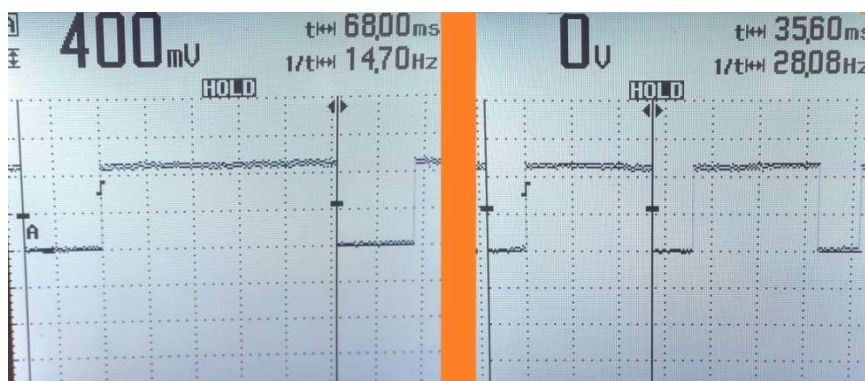
Obrázek 9: Schéma zapojení výkonového členu

Spínání je řešeno přes tranzistor MOSFET-N s typovým označením IRL540N, jehož součástí je logická úroveň. Pomocí logické úrovně je umožněno plné otevření tranzistoru i při malých hodnotách napětí, které se z čipu dostávají. Zde se jedná o napětí, které dosahuje maximální hodnoty okolo 5V. Celý obvod je napájen stejnosměrným napětím 10 V. Výstupní pin z Arduina, podporující pulzní šířkovou modulaci, je připojen na řídicí elektrodu tranzistoru. Řídicí elektroda je pak řízena digitálním binárním signálem z Arduina. Digitální signál je v rozmezí 0 až 255 a odpovídá přepočtu napěťového signálu o velikosti 0 – 10 V. Hodnota digitálního signálu 0 odpovídá napětí 0 V a hodnota digitálního signálu 255 odpovídá napětí 10 V. To znamená, že se změnou hodnoty digi-

tálního signálu se mění výstupní napětí, které je přiváděno na motor, respektive na čerpadlo. Změna digitálního signálu umožňuje ovládat změnu velikost napětí a tak je možné ovládání velikosti (rychlosti) čerpání kapaliny, což umožňuje měnit a následně ovládat velikost průtoku.

2.3 Testování a nastavení výpočetních parametrů

Hned na úvod je třeba si nastavit a parametrizovat výstupní signál z průtokoměru. Výstupním signálem z průtokoměru je obdélníkový signál s frekvencí závislou na velikosti průtoku vody hnané skrz čerpadlo. Výstupní signál z průtokoměru je pak zaznamenán a zobrazen na obrázku 10. Pro záznam a měření výstupního signálu z průtokoměru byl využit přenosný osciloskop od firmy FLUKE s označením FLUKE 124.



Obrázek 10: Průběh výstupního signálu z průtokoměru

Jak lze vidět z obrázku 10, délka periody jednoho pulzu se pohybuje v desítkách milisekund. Od toho se odvíjí i frekvence. V levé části obrázku 10 je nastaven malý průtok. Turbína se točí pomalu a výstupní signál má nízkou hodnotu frekvence. Dojde-li ke zvýšení průtoku, turbína se začne točit rychleji. Následně dojde ke zkrácení doby periody a frekvence se zvýší. Změna frekvence je rovněž vidět z pravé části obrázku 10. Z takto zjištěných a naměřených hodnot se průtok vypočítává následujícím způsobem. Nejprve se stanoví časový úsek nově příchozího pulzu. Z časového úseku se stanoví perioda signálu a následně se vypočte i samotná frekvence. K tomuto účelu

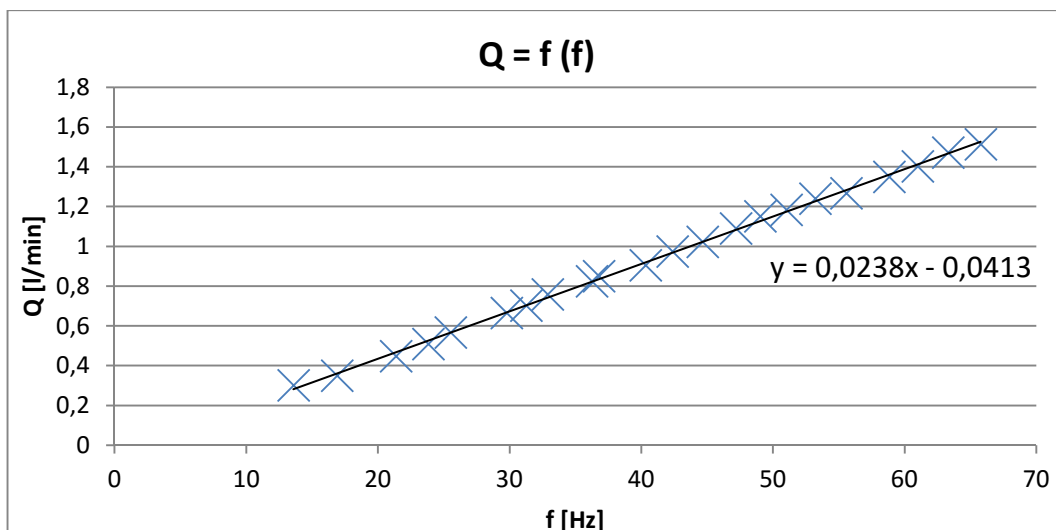
se zde využívá naměřených hodnot průtoků, které jsou zaznamenány v tabulce 1.

Tabulka 1: Naměřené a vypočtené hodnoty průtoků

t [s]	Q [l/s]	Q [l/min]	f [Hz]		t [s]	Q [l/s]	Q [l/min]	f [Hz]
99,7	0,01	0,30	13,58		30,8	0,02	0,97	42,37
85,8	0,01	0,35	16,89		29,4	0,02	1,02	44,64
67,1	0,01	0,45	21,36		27,5	0,02	1,09	47,17
58,9	0,01	0,51	23,8		26,1	0,02	1,15	49,02
53,0	0,01	0,57	25,51		25,3	0,02	1,18	51,02
45,1	0,01	0,67	29,76		24,2	0,02	1,24	53,19
42,8	0,01	0,70	31,25		23,7	0,02	1,27	55,55
39,6	0,01	0,76	32,89		22,3	0,02	1,35	58,82
36,4	0,01	0,82	36,23		21,4	0,02	1,40	60,97
35,3	0,01	0,85	36,76		20,4	0,02	1,47	63,29
33,2	0,02	0,90	40,32		19,8	0,03	1,51	65,78

Naměřené hodnoty z tabulky 1 se následně vynesly do grafu. Výsledkem je získání závislosti průtoků na frekvenci dle grafu 1. Závislost není zcela lineární, což může být ovlivněno nepřesnostmi při měření a výstupním signálem z průtokoměru. Časy uvedené v tabulce 1 představují příslušný časový úsek, za který se přečerpá voda o objemu 0,5 l. Pomocí jednoduché trojčlenky se dostane velikost průtoků, který je v litrech za sekundu. Následným přepočtem se stanoví výsledný průtok v litrech za minutu.

U průtokoměru byla prokázána nepřesnost měření, která se nejvíce projevuje právě u malých průtoků. Při konstantním průtoku dochází ke změně délky pulzů generovaných průtokoměrem, což má za následek odlišné doby period a potažmo i odlišné frekvence. S odlišnými frekvencemi se liší i stanovený průtok. U malých průtoků, které se pohybují přibližně okolo 0,6 l/min a níže, se jedná o rozdíl přibližně $\pm 0,01$ l/min. U větších průtoků, které jsou zhruba nad 0,8 l/min, je rozdíl v průtoku již méně patrný.



Graf 1: Závislost průtoku na frekvenci

Z hodnot vynesných do grafu se dostane již výše zmíněná závislost. Jednotlivé body jsou proloženy spojnici, z níž je vypočtena rovnice grafu. Díky rovnici, která je získána z metody popsané výše, je možné vypočítat průtok z frekvence. Do rovnice se za parametr x dosadí aktuální stanovená frekvence, čímž se získá aktuální hodnota průtoku, která je v rovnici z grafu 1 označena jako y .

Celý výpočetní postup je následně převeden do programovacího jazyka a nahrán do Arduina. Převodem se bude podrobněji zabývat kapitola řešící programovou část celé práce, kde je řešeno stanovení časového úseku jednoho pulzu a periody. Následuje stanovení frekvence a výpočet průtoku. Vše je odzkoušeno a otestováno nejprve v testovacích podmínkách. Následuje opětovné odzkoušení v laboratoři s celou soustavou a ověřením funkčnosti i zde. Bohužel není k dispozici cejchované měřidlo průtoku, podle kterého by bylo možné určit míru přesnosti, nebo nepřesnosti výše popsaného způsobu určeného ke stanovení průtoku. Zároveň by bylo možné stanovit míru přesnosti samotného průtokoměru.

2.4 Návrh softwaru pro regulaci průtoku

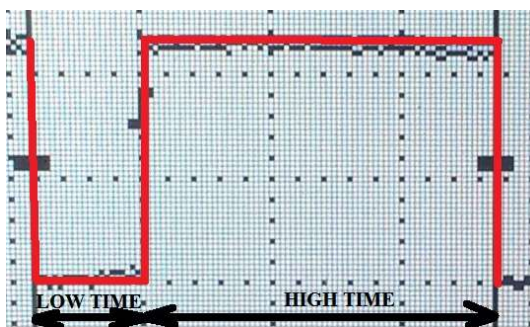
Nejprve je třeba stanovit velikost okamžitého průtoku. Vychází se z kapitoly 2.3 *Testování a nastavení výpočetních parametrů*. Následuje převod výpočetních operací do programové podoby. Výstupní data z průtokoměru jsou přiváděna na příslušný Arduino

pin, konkrétně jde o digitální pin 8. Prvním krokem je stanovení délky jednoho pulzu pomocí zdrojového kódu 1.

```
LowTime = pulseIn(Tokmer, LOW);  
HighTime = pulseIn(Tokmer, HIGH);  
Perioda = HighTime + LowTime; // Doba jednoho pulzu
```

Zdrojový kód 1: Stanovení doby jednoho pulzu

Řešení je na třech řádcích kódu. Pro názornější představu, jak zdrojový kód 1 funguje, je zde přiložen ilustrační obrázek 11, kde je vidět stanovení délky jednoho pulzu. Využívá se reakce na náběžnou a sestupnou hranu. Nejprve se zaznamená časový úsek, který je označený jako *LOW TIME*, první řádek zdrojového kódu 1. Následně se zaznamená časový úsek pulzu, který je označen jako *HIGH TIME*, což je druhý řádek zdrojového kódu 1. Oba časové úseky jsou sečteny na třetím řádku kódu, který reprezentuje periodu příchozího signálu z průtokoměru. Hodnoty *LOW* a *HIGH* jsou způsobeny tzv. pull up rezistorem, který je připojen mezi piny průtokoměru zobrazeného na obrázku 6. Perioda je počítána pro každý nově příchozí pulz. Tak se získává vždy aktuální hodnota periody.



Obrázek 11: Časový úsek jednoho pulzu

Ze stanovené periody pro jeden pulz se následně určí frekvence s využitím základního vzorce $f = 1/T$. Posledním krokem je výpočet průtoku ze stanovené frekvence. Pro výpočet se využije rovnice z grafu 1, která se přepíše do tvaru $Q = 0,0238 * f - 0,0413$.

```
T = Perioda*0.001;    ///Perioda [s]
F = 1/(T*0.001);      //Frekvence [Hz]
Q = 0.0238*F-0.0413; //Prutok [L/min]
```

Zdrojový kód 2: Výpočet periody, frekvence a průtoku

Výše popsané výpočetní operace jsou opět realizovány pomocí třech řádků v podobě zdrojového kódu 2, kde se jedná o využití dvou matematických vzorců spolu s převodem na základní jednotky a následným výpočtem průtoku, který je udávám v litrech za minutu. Zároveň je zdrojový kód 1 a zdrojový kód 2 základem pro regulaci průtoku a následné řízení více rozměrového systému se dvěma vstupy a dvěma výstupy. Zde bude rovněž využito výše popsané metody pro stanovení aktuální hodnoty průtoku.

Nyní je Arduino schopné vypočítat a stanovit velikost průtoku vody z výstupních dat získaných z průtokoměru. Stanovení periody, výpočet frekvence a průtoku je opět prakticky odzkoušeno a porovnáno s hodnotami naměřenými osciloskopem (perioda a frekvence signálu).

2.4.1 Regulátor průtoku

Využívá se číslicová verze PID regulátor. PID regulátor je tvořen třemi základními složkami - proporcionální, integrační a derivační. Složky představují tři různé zásahy v reakci na regulační odchylku. Jednotlivé parametry regulátoru mají svůj význam a dají se libovolně nastavovat tak, aby se docílilo požadavků, které jsou kladeny na kvalitu regulace. Do regulátoru vstupuje regulační odchylka a výstup je v podobě váženého součtu výše zmiňovaných složek v podobě akčního zásahu.

Algoritmus PID regulátoru vychází z ideálního paralelního tvaru v časové oblasti, označovaný také jako standartní tvar rovnice 1. V praxi se tento tvar regulátoru využívá přibližně v 31% průmyslových regulátorů. Parametr r_0 představuje proporcionální složku, T_i je integrační časová konstanta a T_d je derivační časová konstanta [8].

$$u(t) = r_0 \left[e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right] \quad (1)$$

První složkou regulátoru je proporcionální člen, což je v podstatě násobení konstantou. Nejprve se stanoví regulační odchylka $e(t)$. Odchylka je dána jako rozdíl hodnoty nastavené a regulované veličiny. Následně se regulační odchylka vynásobí s proporcionální složkou. Tak se získává již výše zmíněný proporcionální člen. Právě zmíněné výpočetní operace se zapíše do podoby, ve které je zdrojový kód 3. Kód je řešen pomocí dvou řádků. První řádek je výpočet regulační odchylky a na druhém řádku se provede výpočet proporcionální složky.

```
double error = mySetpoint - input;
double PTerm = kp * error;
```

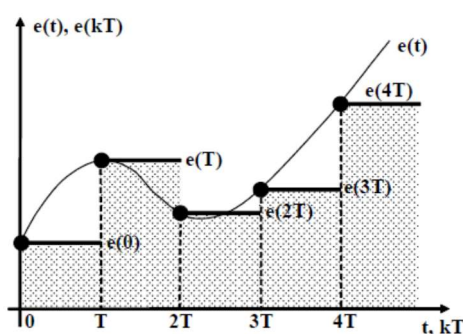
Zdrojový kód 3: Výpočet proporcionálního členu

Následuje integrační složka regulátoru. Nejprve je nutné si uvést geometrický význam určitého integrálu, který má podobu $\int_0^t e(\tau) d\tau T_d$. Jedná se o plochu obrazce shora ohraničeného integrovanou funkcí $e(\tau)$. Zdola ohraničeného časovou osou a po stranách ohraničeného přímkami $\tau = 0$ a $\tau = t$. Hodnoty regulační odchylky jsou známy pouze v okamžicích vzorkování nikoli průběžně, což znemožňuje přesné zjištění této plochy. Proto je nutné ji aproximovat v podobě součtu obsahů jednoduchých elementárních obrazců - obdélníků či lichoběžníků. Z čehož vyplývá možnost využití dvou metod [8].

a) Levá obdélníková metoda

První metoda pro výpočet integrační složky je obdélníková, přesněji se jedná o tzv. Levou obdélníkovou metodu. Ta je popsána vztahem zobrazeného v podobě rovnice 2.

$$\frac{r_0}{T_i} \int_0^t e(\tau) d\tau \approx \frac{r_0 T_v}{T_i} \sum_{i=1}^k e(i) = I_{OL}(k) \quad (2)$$



Obrázek 12: Levá obdélníková metoda

Hodnota integrálu se nahradí součtem ploch obdélníků nahrazujícím plochu pod původní spojitou křivkou $e(t)$. Vzorkovací perioda pak určuje šířku obdélníku. Aktuální hodnota vzorku $e(iT)$ v čase $t = iT$, kde $i = 1, \dots, k$, určuje výšku i -tého obdélníku.

Řešení levé obdélníkové metody je programově řešeno na prvním řádku zdrojového kódu 4. Zde je vidět jednoduché řešení této metody. Zároveň je součástí programový kód na potlačení takzvaného wind-up efektu. Wind-up efekt způsobuje integrační složka. Akční veličina je vždy omezená a integrační složka je omezena pouze maximální hodnotou používané v aritmetice v počítači. Čili se jedná o prakticky neomezenou veličinu, proto se využívá dynamického omezení integrační složky. Leží-li akční veličina mimo realizovatelnou oblast, použije se minulé hodnota integrační složky a nikoli aktuální. Dojde ke zmrazení integrační složky po dobu, kdy je akční veličina mimo rozsah do doby, než bude opět akční veličina v realizovatelné oblasti, což řeší druhý a třetí řádek kódu ve zdrojovém kódu 4.

```

ITerm+= (ki * error);

if(ITerm > outMax) ITerm= outMax;

else if(ITerm < outMin) ITerm= outMin;

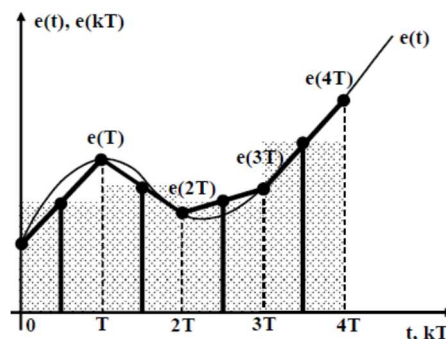
```

Zdrojový kód 4: Programové řešení integrační složky (Obdélníková metoda)

b) Lichoběžníková metoda

Druhou metodou pro zjištění integrační složky je tzv. lichoběžníková metoda. Zde se nahrazuje hodnota integrálu součtem ploch lichoběžníků nahrazujícím plochu opět pod původní spojitou křivkou $e(t)$. Jde o přesnější aproximaci spojitě křivky $e(t)$, než v případě předchozím. Lichoběžníková metoda je v programu řešena v podobě zdrojového kódu 5.

$$\frac{r_0}{T_i} \int_0^t e(\tau) d\tau \approx \frac{r_0 T_v}{2 T_i} \sum_{i=1}^k (e(i) + e(i-1)) = I_L(k) \quad (3)$$



Obrázek 13: Lichoběžníková metoda

Opět na prvním řádku je proveden výpočet integrační složky, která vychází ze vztahu zobrazeného v rovnici (3). Vztah pro výpočet integrační složky je v podobě rekurzivně počítané diferenční rovnice a to z toho důvodu, že počítání sumy vždy od začátku, by bylo nepraktické a celkem nerozumné. I zde je ošetřena integrační složka proti wind-up efektu, druhý a třetí řádek zdrojového kódu 5.


```
double ITerm += lastITerm + 0.5 * kp * SampleTime * (error + lastError) / ki;

if(ITerm > outMax) ITerm= outMax;

else if(ITerm < outMin) ITerm= outMin
```

Zdrojový kód 5: Programové řešení integrační složky (Lichoběžníková metoda)

Třetí a poslední složkou tohoto regulátoru je derivační složka. Nejjednodušší variantou je náhrada první diferencí, která je v následujícím tvaru v podobě rovnice 4.

$$r_0 T_d \frac{de}{dt} \approx \frac{r_0 T_d (e(k) - e(k-1))}{T_v} = D(k) \quad (4)$$

Bohužel, nevýhodou této jednoduché náhrady je její citlivost na šum. Proto se spíše využívá vhodnější metoda filtrované derivace, při použití Tustinovy metody, která je v podobě diferenční rovnice zobrazena v rovnici 5 [8].

$$d(k) = \frac{2r_0 T_d}{(T_v + 2T_d \alpha)} (e(k) - e(k-1)) + \frac{(2r_0 T_d - T_v)}{(T_v + 2T_d \alpha)} d(k-1) \quad (5)$$

Programové řešení derivační složky je pak možné využít ve dvou tvarech. První tvar zdrojového kódu 6 je rozdílem dvou hodnot a to konkrétně aktuální vstupní hodnoty a předchozí vstupní hodnoty.

```
double dInput = (input - lastInput);

double DTerm = kd * dInput;
```

Zdrojový kód 6: Programové řešení derivační složky

Programově je zde řešen i druhý zmiňovaný tvar sloužící ke stanovení derivační složky, který vychází z rovnice 5. Jeho programové řešení je vidět ve zdrojovém kódu 7.

```
double alfa = 0.09; //volba v rozmezi (0.05 - 0.2)

double ad1 = 2 * kp * kd/(sampleTime + 2 * kd * alfa);

double ad2 = (2 * kd * alfa - sampleTime)/(sampleTime + 2 * kd *
alfa);

double DTerm = ad1 * (error - lasterror) + ad2 * lasterror;
```

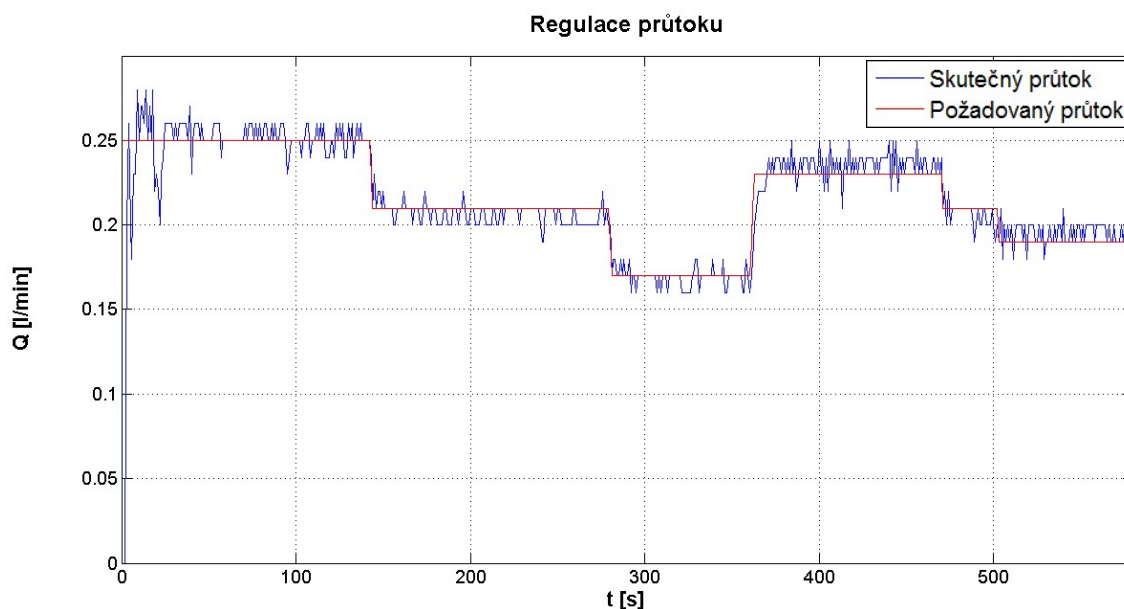
Zdrojový kód 7: Programové řešení derivační složky (filtrovaná derivace)

Programová část tak umožňuje volbu metody, pomocí níž je možné stanovení jednotlivých složek. Ovšem pro modifikaci programu, která je zde aktuálně využívána, a která se více osvědčila, je obdélníková metoda spolu s filtrovanou derivací. Po stanovení jednotlivých složek regulátoru, respektive jejich výpočtu, přijde na řadu součet všech třech složek. Jedná se o složky proporcionální, integrační a derivační. Výsledný součet se provádí v proměnné s názvem *Output*, reprezentující výstup z regulátoru, který je následně přiveden na výkonový člen prvku, který má být regulován. Výstupní hodnota *Output* nabývá velikosti digitálního signálu v rozmezí 0 až 255. Přičemž digitální hodnota 255 odpovídá maximálnímu napětí 12 V a digitální hodnota 0 odpovídá minimu, tedy hodnotě napětí 0 V.

Celé programové řešení regulace je ve formě knihovny, která je nadefinována a implementována hned na začátku programu. Dále je nutné nadefinovat několik parametrů, s kterými bude následně regulátor pracovat. Jedná se o vstupní, výstupní a žádané hodnoty spolu s nastavením proporcionální, integrační a derivační složky. Ty jsou přednastaveny již na začátku programu. Samozřejmostí je i možnost měnit tyto parametry dle požadavků uživatele.

Součástí regulátoru průtoku je navíc ještě výstupní napěťový signál, který je závislý na aktuálním průtoku. Napěťový signál je vyveden na dvojici svorek, na kterých bude možné signál měřit a následně jej využívat pro další zpracování. Napěťový signál je závislý na velikosti průtoku. S rostoucím průtokem narůstá i napěťový signál a obráceně, tedy s klesajícím průtokem dochází ke snížení napěťového signálu. Velikost napě-

řového signálu je opět v rozmezí 0 – 10 V. Mezní hodnoty průtoků se pohybují okolo 0,16 – 0,30 l/min, čemuž odpovídá napěťový signál, který se pohybuje přibližně v rozmezí 2 – 6 V. Z čehož vyplývá, že výstupní napěťový signál, který závisí na průtoku, se pohybuje ve výše zmíněném rozmezí.



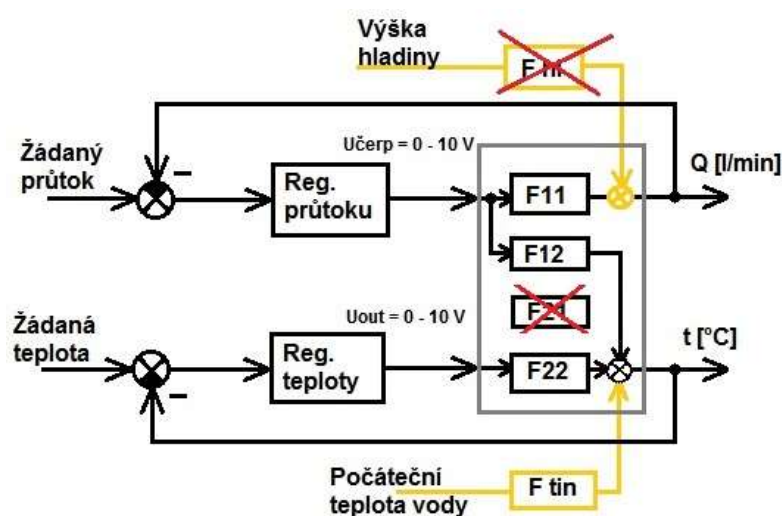
Graf 2: Ověření funkčnosti regulace průtoku

Na závěr této kapitoly je třeba ještě provést ověření funkčnosti regulátoru průtoku. Z dat získaných při experimentálním měření se vytvořil graf zobrazující průběh regulace průtoku, který je vidět z grafu 2. Průtok je regulován pomocí PI regulátoru s parametry $P_T = 1,60$ a $I_T = 0,02$. Volba parametrů vychází z modelu, který se nastavoval pomocí autotuningu a ručního nastavení. Experimentálním měřením byla prokázána funkčnost regulátoru průtoku, který je nyní připraven pro další manipulaci a práci s regulátorem.

3 Regulace průtoku a teploty

3.1 Popis systému

Celá soustava je nyní reprezentována systémem, který má dva vstupy a dva výstupy. Blokové schéma tvoří dvě regulační smyčky spolu s nežádoucími vlivy, které působí více či méně negativně na regulační pochody. Základní princip regulace průtoku a teploty je vidět na obrázku 14. Z blokového schématu lze vidět, že se jedná o dva regulátory, jejichž regulační pochody se provádějí současně. Prvním regulátorem se reguluje velikost průtoku a druhým regulátorem se reguluje teplota vody. Regulátor teploty je zrealizován stejným způsobem, kterým je zrealizován i regulátor průtoku, tedy programově pomocí platformy Arduino. Aktuální teplota je snímána pomocí teplotního čidla a přivádí se na vstup regulátoru teploty spolu s žádanou hodnotou teploty. Výstupní signál z regulátoru je přiveden na vstup napěťově řízeného zdroje napětí (0 V – 10 V / 0 V – 230 V) na jehož výstupu je připojeno topné těleso.



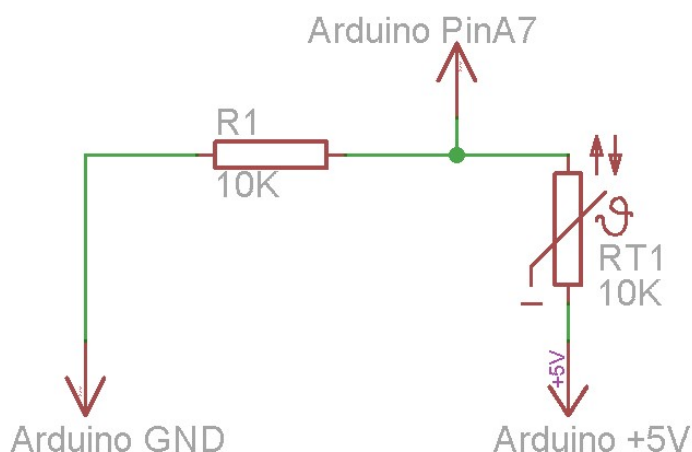
Obrázek 14: Blokové schéma regulace teploty a průtoku

Z blokového schématu je dále vidět takzvaná křížová vazba. Změna regulačního pochodu jedné smyčky ovlivňuje svým akčním zásahem i regulační pochod druhé smyčky. Jedná se o blok přenosové funkce s označením F_{12} . Druhá křížová vazba zde není, jak je patrné z obrázku 14, jelikož změnou výkonu topného tělesa nedochází k ovlivnění veli-

kosti průtoku. Tím pádem není třeba druhou křížovou vazbu brát v úvahu. Dalším vlivem, který zde je, je změna výšky hladiny vody v nádobě, ve které je umístěno čerpadlo. Vliv změny výšky hladiny není zanedbatelný, proto je snaha tento vliv eliminovat právě pomocí regulátoru průtoku. Ještě je třeba zmínit vliv v podobě teploty vody přitékající do válce s topným tělesem. Přitékající voda má zpravidla pokojovou teplotu, tedy teplotu okolo 18 °C. Při běžném provozu se tento vliv téměř neprojeví a je tedy nepatrný. Z tohoto důvodu je možné teplotní vliv zcela zanedbat.

3.2 Snímání a regulace teploty

Před samotnou regulací teploty je nejprve nutné zvolit vhodný způsob, pomocí něhož bude možné měřit teplotu vody v nádobě. K měření teploty se zde využívá NTC termistor s odporovou hodnotou 10 k Ω . Vlastností, která je charakteristická pro tento typ součástky, je změna odporové hodnoty se změnou teploty, tudíž odpor je závislý na teplotě. U NTC termistoru klesá jeho odporová hodnota s rostoucí teplotou a naopak. Schéma zapojení není nijak složité, což lze vidět níže z obrázku 15.



Obrázek 15: Schéma zapojení termistoru NTC

Na analogový pin, Arduino pinA7, se přivádí signál, který se mění s teplotou. Ovšem signál je třeba ještě převést, respektive přepočítat na požadovanou teplotu. Jelikož je třeba zobrazovat aktuální hodnotu teploty, využívá se zde tzv. Steinhart-Hart rovnice [13], která má tvar v podobě rovnice 7, což umožní aproximaci přepočtu hodnoty odpo-

ru na teplotu. Nejprve je nutné provést ještě jeden výpočet, s jehož pomocí je možné stanovit aktuální hodnotu na termistoru. Výpočet se provede za pomoci rovnice 6, kde za R_{term} se dosadí odporová hodnota termistoru, která je uvedena ve schématu na obrázku 15. Za proměnnou V_{out} se dosadí výstupní napětí na desce, které je v tomto případě reprezentováno digitálním signálem, jehož hodnota je v rozmezí 0 – 1023. Hodnota je získávána z Arduino PinA7, kam je připojen termistor. Po zjištění hodnoty na termistoru, už zbývá jen zahrnout tuto hodnotu do již výše zmíněné Steinhart- Hartovi rovnice 7.

$$R = R_{term} * \left\{ \left(\frac{1024,0}{V_{out}} \right) - 1 \right\} \quad (6)$$

$$T = \frac{1}{A} + B * \{[\ln(R)] + C * [\ln(R)]^3\} \quad (7)$$

Parametry A , B a C jsou konstantní parametry, které jsou uvedeny v tabulce 2 [13]. Uvedené koeficienty jsou pro termistor s odporovou hodnotou 10 k Ω . Výsledkem rovnice je pak aktuální teplota v Kelvinech, proto je zde ještě nutný převod teploty z Kelvinů na stupně Celsia.

Tabulka 2: Hodnoty konstant pro Steinhart- Hart rovnici

A	0,001129148
B	0,000234125
C	8,767410000E-08

Výše popsané výpočetní operace a rovnice, je nyní třeba převést do programového jazyka a následně naimplementovat do Arduina. Výsledné programové řešení je vidět v podobě zdrojového kódu 8, který je výřezem z celého programu.

```
double Thermistor(int RawADC) {
double Temp;

Temp = log(10000.0*((1024.0/RawADC)-1)); //Odpor termistoru
Temp=1/(0.001129148+(0.000234125+(0.0000000876741*Temp*Temp))*Temp);
Temp = Temp - 273.15; // Převod Kelvin na °C
return Temp; }

double Teplota = (int(Thermistor(analogRead(7))));
```

Zdrojový kód 8: Výpočet teploty pomocí Steinhart- Hart rovnice

Celý výpočet teploty je ve funkci, která je volána v případě, že chceme měřit teplotu. Na třetím řádku zdrojového kódu 8 se vypočítává aktuální hodnota odporu na termistoru. Na následujícím řádku se provádí výpočet dle Steinhart-Hart rovnice, viz rovnice 7. Následný převod teploty z Kelvinů na stupně Celsia se provede na pátém řádku zdrojového kódu. Z posledního řádku zdrojového kódu 8 je vidět, že termistor je připojen k Arduino přes analogový vstup s číselným označením sedm (Arduino PinA7). Ve chvíli, kdy je zavolána funkce *TeplotniCidlo()*, v níž je uložena právě proměnná *Teplota*, je možné sledovat aktuální teplotu, která je měřena teplotním čidlem zobrazeným na obrázku 16. Na obrázku 16 je vidět finální řešení teplotního čidla (pravá část obrázku). Jak je z obrázku patrné, na konci kabelu je kovový hrot, který byl původně dutý. Uvnitř kovového hrotu se nachází výše zmíněný termistor (levé část obrázku 16). Celé je to zalité nevodivou hmotou a opatřeno ještě smršťovací bužírkou, která brání vniknutí kapaliny dovnitř hrotu.

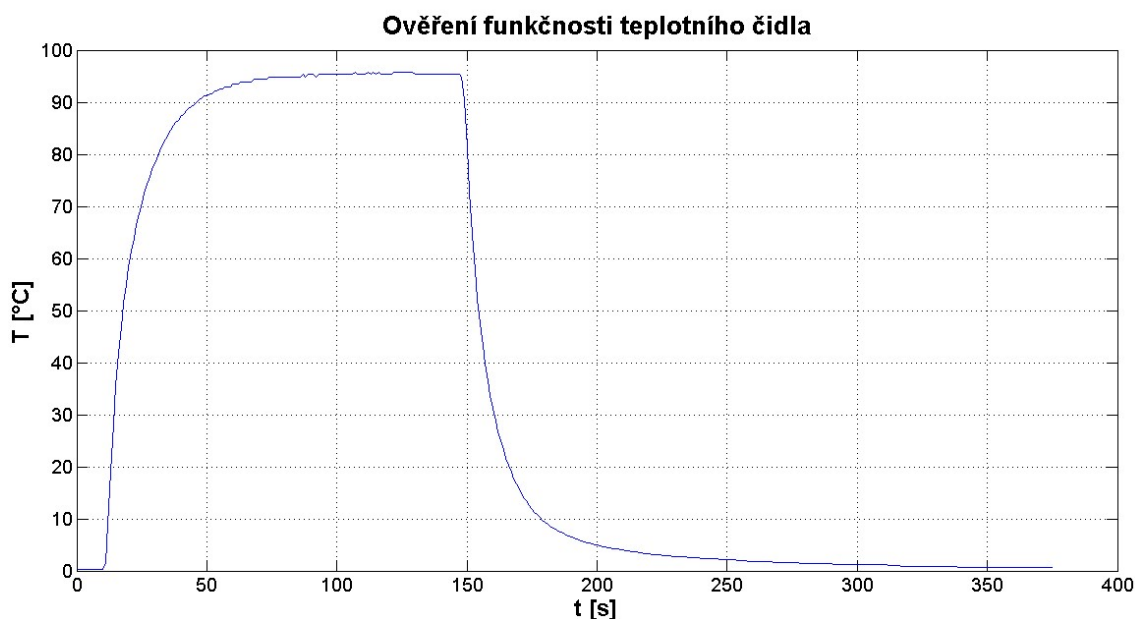


Obrázek 16: Termistor a teplotní čidlo

Na závěr je nutné provést ověření funkčnosti teplotního čidla a softwarové části zajišťující výpočet a stanovení aktuální teploty. Aktuální teplota se během ověřování vypisuje do sériového monitoru v programu Arduino a současně se aktuální hodnota teploty zobrazuje na displeji. Ověřovací proces se provádí vložení teplotního čidla do vody s ledem. Naměřená teplota by měla odpovídat 0 °C. Druhým krokem je vložení čidla do vroucí vody. Zde by měla být naměřena teplota přibližně 100 °C a následné opětovné ponoření do ledové vody o teplotě 0 °C. Z hodnot, které byly naměřeny během funkčních testů, vznikla tabulka 3. Výsledný průběh je vyneseno do grafu 3.

Tabulka 3: Naměřené hodnoty teplot a odporů při funkčních testech

Prostředí	t [°C]	R [kΩ]
Led	1	29,91
Místnost	25	9,92
Konvice	97	0,71

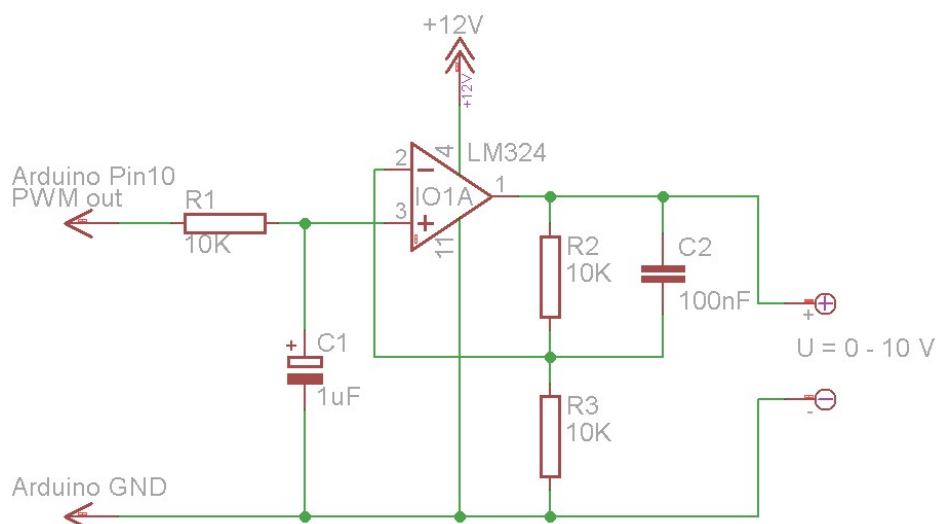


Graf 3: Průběh teploty při ověřování funkčnosti teplotního čidla

Do nádoby s vodou a ledem se ponoří čidlo a provede se první měření teploty. Výsledné teplotě 1 °C pak odpovídá odporová hodnota 29,91 kΩ. Následně se čidlo ponoří do vroucí vody a opět se provede měření teploty. Změřené teplotě 97 °C odpovídá odporová hodnota 0,71 Ω. Posledním bodem je měření teploty při pokojové teplotě, která byla

25 °C. Odporová hodnota při pokojové teplotě je 9,92 k Ω . Z naměřených dat získaných při funkčních testech teplotního čidla a programové části vznikl graf 3, kde je vidět popsaná změna teplot z 1 °C na 97 °C a následná změna z 97 °C na 1 °C. Současně s funkčním měřením teploty NTC termistorem byla ještě teplota měřena cejkovaným rtuťovým teploměrem, aby bylo možné ověřit funkčnost a přesnost stanovené teploty. Použitý kontrolní teploměr má rozlišení 0,1 °C a uvedená přesnost měření je ± 0.5 °C. Dále při porovnání naměřených odporových hodnot s odporovými hodnotami, které jsou uvedeny v katalogovém listu, je vidět, že se od sebe příliš neliší. Funkčními testy se zjistilo, že teplotní čidlo je plně funkční a přesnost měřené teploty se pohybuje v rozmezí ± 0.5 °C. Výrobce v katalogových listech určených pro termistor NTC-640 10 k Ω udává teplotní odchylku ± 1 %. Navíc v nich výrobce uvádí jednotlivé hodnoty teplot spolu s odpovídajícími hodnotami odporů.

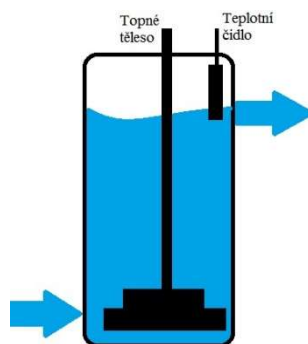
Nyní je zařízení schopno měřit aktuální teplotu vody v nádobě s topným tělesem, proto se může přejít k samotné regulaci. Značnou výhodou je zde fakt, že samotný regulátor je již připraven a naprogramován z předešlé kapitoly, kde se prováděla regulace průtoku, čímž se ušetří část práce. Je pouze potřeba vhodně modifikovat regulátor tak, aby umožnil regulaci teploty namísto průtoku. Vstupem do regulátoru je požadovaná hodnota teploty a aktuální teplota ve válcové nádobě s vodou a topným tělesem. Výstupem z regulátoru je opět digitální signál v rozmezí 0 až 255, který se zapisuje na Arduino pin 10. Výstup z pinu je přiveden na vstup RC článku, který je na vstupu schématu na obrázku 17, čímž se dosáhne vyhlazení výstupního signálu z Arduino pinu 10, který je ve formě PWM signálu. Vyhlazený signál je ještě zesílen pomocí zesilovače s označením LM324, který je zapojený jako neinvertující zesilovač. Výstupní napětí ze zesilovače z obrázku 17 je v rozmezí od 0 V až do 10 V v závislosti na velikosti digitálního signálu z regulátoru, potažmo z Arduina. K výstupu z převodníku se připojí napětově řízený zdroj napětí. Ten zajišťuje převod stejnosměrného vstupního napětí 0 – 10 V na výstupní střídavé napětí v rozmezí 0 – 230 V, kterým se napájí topné těleso, tudíž změnou výstupního signálu z Arduina se dá ovládat výkon topného tělesa.



Obrázek 17: Schéma převodníku z PWM signálu na $U_{dc} = 0 - 10 \text{ V}$

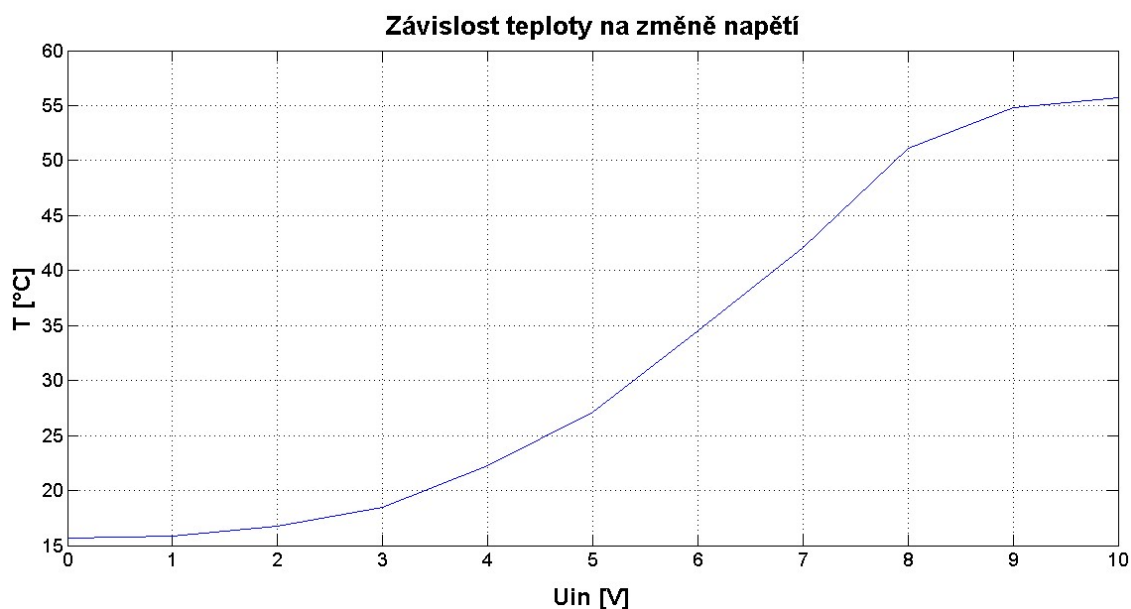
3.3 Vliv průtoku na teplotu

Základním principem jsou zde dva regulátory, kdy jeden regulátor řídí pouze jednu výstupní veličinu. Situace pak vypadá následovně. Prvním regulátorem se provádí regulace průtoku a druhým regulátorem se provádí regulace teploty vody. V tomto případě se jedná o otevřenou vodní soustavu, jak již bylo řečeno v přechozí kapitole. Voda tedy vytéká pryč a nedochází tak k návratu ohřáté vody zpět do soustavy, proto je nutné, pro chod celého procesu kontrolovat a následně doplňovat vodu do nádoby s čerpadlem. Teplota čerpané vody z nádoby se pohybuje okolo 18°C , což odpovídá okolní teplotě v laboratoři. Na obrázku 18 je vyznačen přítok a odtok vody ve válci a dále umístění topného tělesa a teplotního čidla. Přitékající chladnější voda se drží ve spodní části válce a naopak teplejší voda se drží v horní části. Při standardním operačním režimu se na teplotě vody ve válci utvoří profil, kdy teplota vody stoupá s výškou. Měřená teplota odtékající vody se pak může považovat za téměř konstantní. Problém nastává ve chvíli, kdy je nastaven nízký průtok vody a vysoký výkon topného tělesa. Pro tento případ by bylo vhodnější vodu ve válci míchat ručně, nebo mechanicky, čímž by se docílilo opět konstantní teploty ve válci. Oblast, která je pro regulaci průtoku vymezena rotametrem, není postihována uvedeným problémem, proto není třeba zajišťovat míchání vody.



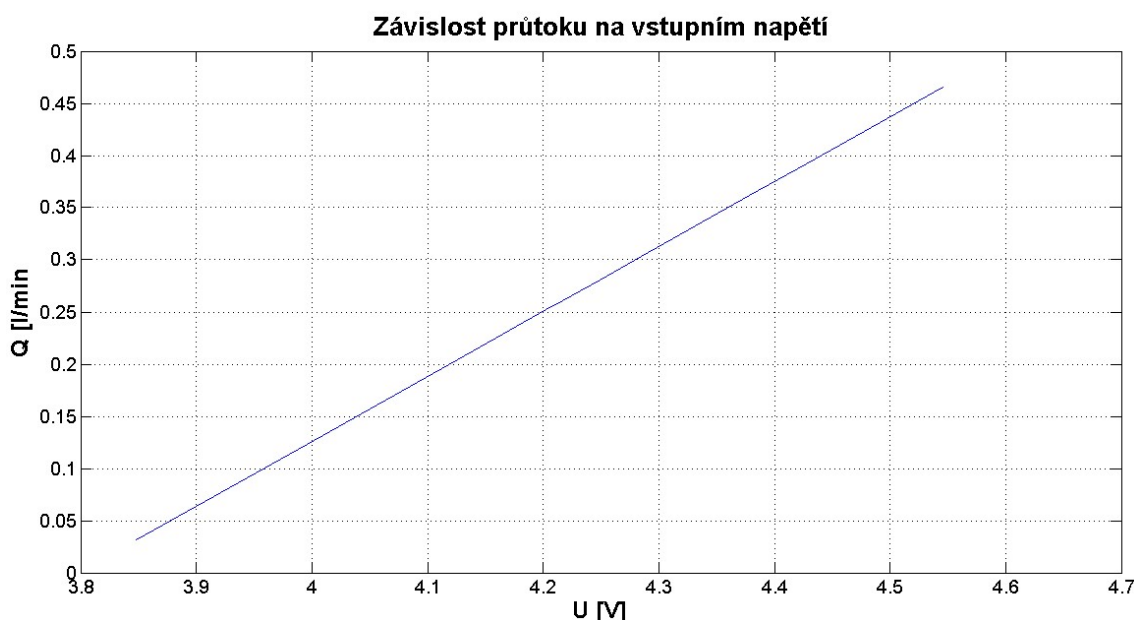
Obrázek 18: Nádob s vodou, topným tělesem a teplotním čidlem

Ovšem takto realizovaný systém v sobě ukrývá jednu podstatnou nectnost, jejíž existenci prokáže následující experimentální měření, následované metodami, které pomáhají tuto nectnost potlačit a eliminovat. Nejprve je třeba proměřit závislost teploty na změně vstupního napětí přivedeného na napěťově řízený zdroj napětí. Výsledná závislost je vynesena do grafu 4, z kterého je vidět, že výsledná závislost není lineární. Ovšem na intervalu vstupního napětí v rozmezí od 4,5 V až do 8 V je závislost téměř lineární a současně se jedná o oblast, ve které se soustava během provozu používá. Ještě je třeba dodat, že naměřená statická charakteristika z grafu 4, je měřena při průtoku 0,21 l/min.



Graf 4: Závislost teploty na změně vstupního napětí

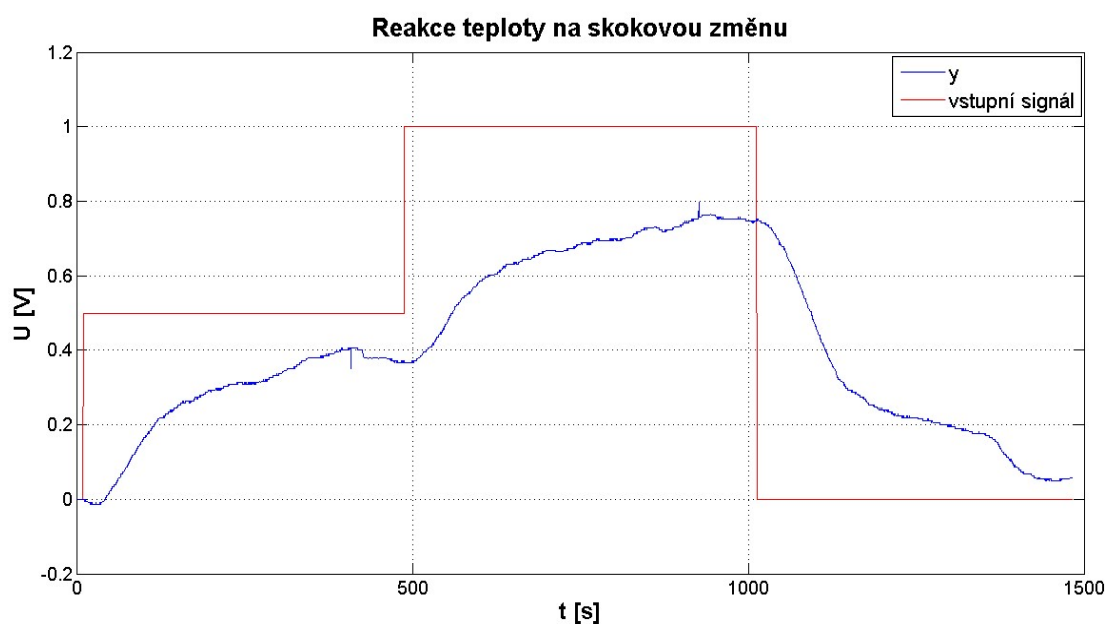
Stejný postup je i při stanovení závislosti průtoku na vstupním napětí na čerpadle, který se mění téměř lineárně v celém svém rozsahu na rozdíl od teploty, což je patrné z naměřených hodnot, které jsou následně vyneseny do grafu 5. Interval vstupních hodnot je volen mezi minimálním a maximálním požadovaným průtokem, který je vyznačen na rotametu. Vymezený interval, tvořený dílkou rotametu, odpovídá průtoku od 0,15 l/min až do 0,30 l/min, což je interval, ve kterém se průtok pohybuje při standardním provozu. Takto vymezenému intervalu průtoku pak odpovídá vstupní napětí na čerpadle v rozmezí od 4,0 V do 4,4 V. Pomocí linearizace a pracovních bodů je možné stanovit příslušné statické charakteristiky. Ze získaných charakteristik se následně stanoví příslušné přenosové funkce.



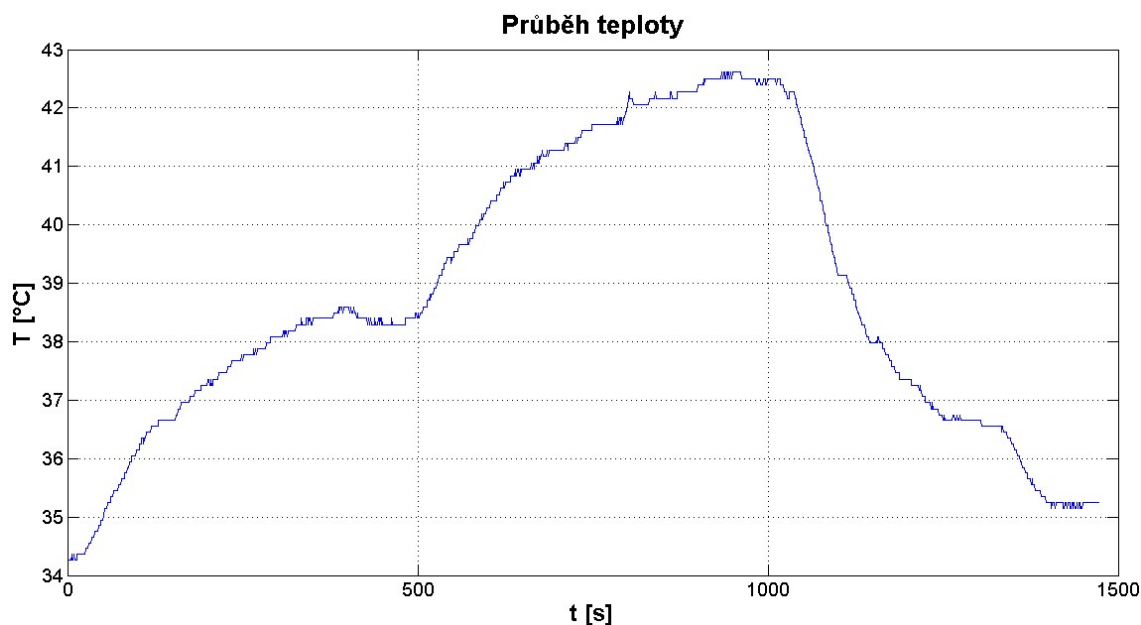
Graf 5: Změna průtoku při změně vstupního napětí na čerpadle

Nejprve se nastaví průtok na konstantní hodnotu. K tomu se zde využívá PI regulátor průtoku realizovaný pomocí Arduina. Dále se nastaví počáteční hodnota vstupního napětí přiváděného na vstup topného tělesa přes napěťově řízený zdroj napětí, která vychází z grafu 4. Po ustálení teploty vody v nádobě s teplotním čidlem se provede první skoková změna vstupního napětí přiváděného na topné těleso přes napěťově řízený zdroj napětí. Výsledkem je změna výkonu topného tělesa a tím i změna teploty vody v nádobě. Další skoková změna je opět v podobě zvýšení vstupního napětí přiváděného

na topné těleso a tím i zvýšení výkonu. Poslední skoková změna je v podobě snížení vstupního napětí zpět na počáteční hodnotu napětí přiváděného na vstup topného tělesa. Před každou skokovou změnou je vždy třeba vyčkat na ustálení teplotních hodnot a poté provést skokovou změnu. Celý experiment je zachycen pomocí třech grafů. Graf 6 zobrazuje výše popsany experiment, kdy se skokově mění napětí, které je přiváděno na vstup topné spirály.

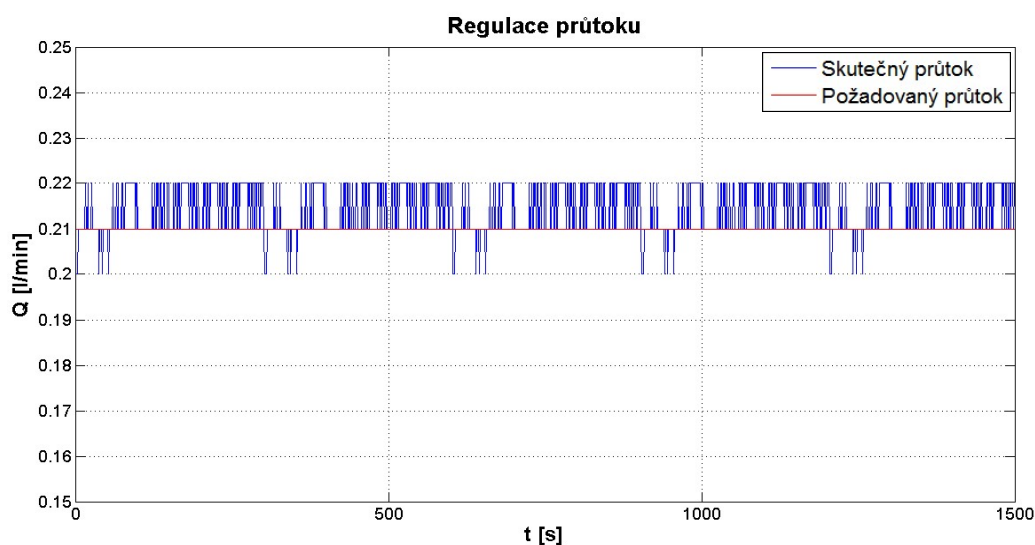


Graf 6: Změna výstupní teploty při skokové změně napětí na vstupu topného tělesa



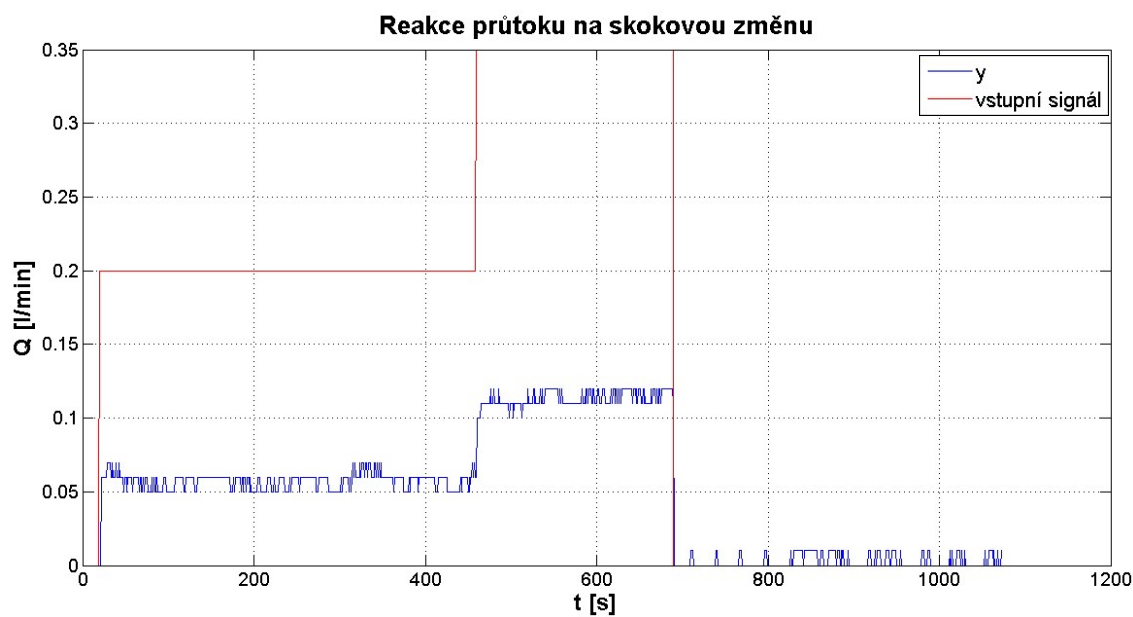
Graf 7: Změna výstupní teploty měřená pomocí teplotního čidla a Arduina

Současně se provádí měření teploty vody v nádobě pomocí Arduina a teplotního čidla. Výsledný průběh měřené teploty je vyneseno do grafu 7. Z grafu je vidět, že průběh teploty měřený teplotním čidlem odpovídá průběhu teploty z grafu 6, který je měřen pomocí teplotního čidla PT100. V poslední řadě z tohoto experimentálního měření vzešel graf 8, který potvrzuje očekávaný výsledek, kdy změna teploty nijak neovlivní změnu průtoku. S rostoucím či klesajícím průběhem teploty nedochází k ovlivňování regulovaného průtoku.

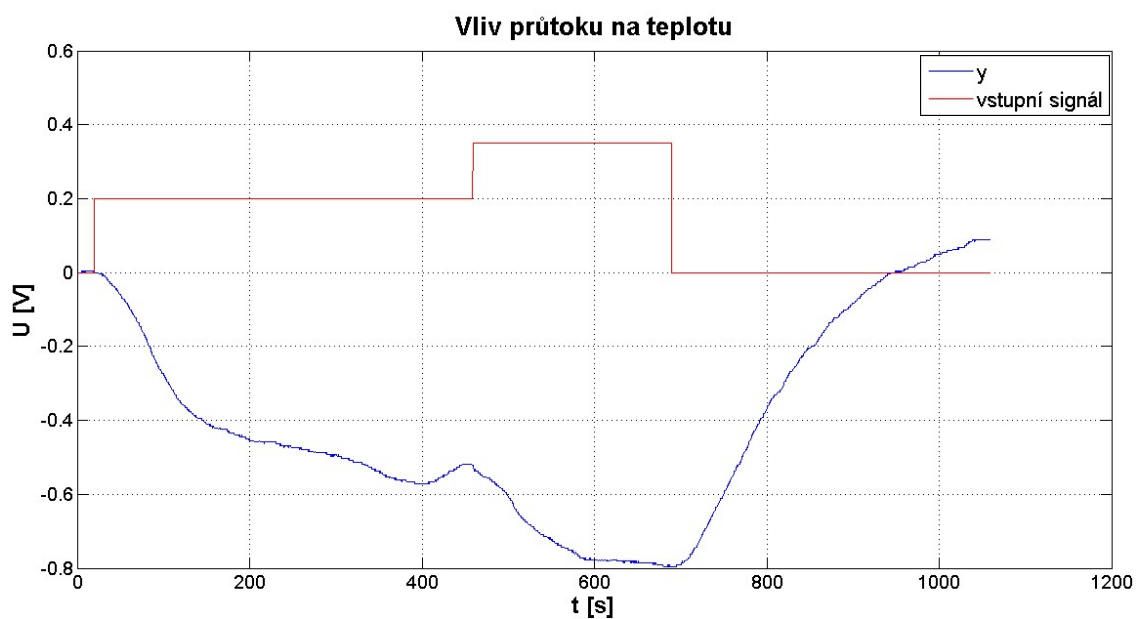


Graf 8: Regulace průtoku pomocí PI regulátoru – nastavení konstantního průtoku

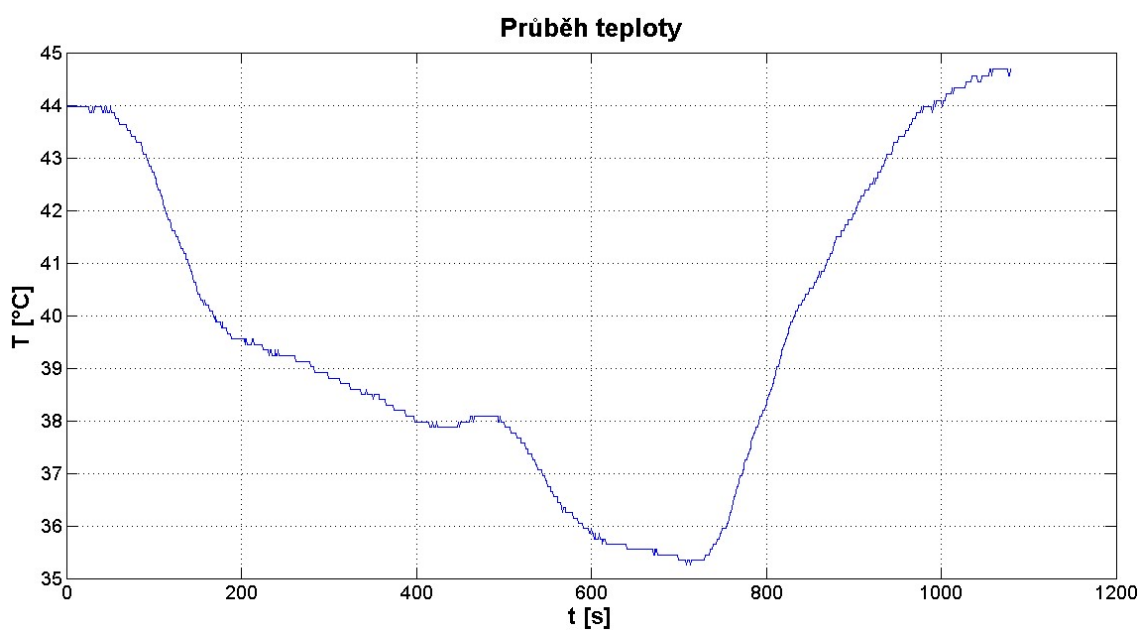
Následuje druhé experimentální měření, při kterém se nastaví konstantní hodnota výkonu topného tělesa. Na vstup topného tělesa se přivede přes napěťově řízený zdroj konstantní hodnota napětí z měřicí karty PCI 1711. Hodnota napětí je tedy pevně nastavena a po čas měření se nemění. Co se během měření mění, je skoková změna vstupního napětí přivedeného na vstup čerpadla, čímž dojde ke změně průtoku. Jednotlivé skokové změny vstupního napětí na čerpadle a jejich vliv na změnu průtoku je zobrazen v grafu 9. Na rozdíl od předešlého experimentálního měření, kde změna teploty neovlivňovala velikost průtoku, tak při tomto experimentálním měření dochází k ovlivňování teploty při změně průtoku. Negativní vliv průtoku na teplotu je zaznamenám a vyneseno do grafu 10. Z grafu je vidět, že s rostoucí změnou průtoku klesá teplota v nádobě s topným tělesem a naopak, čímž je prokázána existence jedné křížové vazby. I zde byla teplota měřena teplotním čidlem. Naměřený průběh teploty je v grafu 11. Výsledný průběh opět kopíruje průběh teploty z grafu 10, kde se teplota zaznamenávala pomocí teplotního čidla PT100, stejně jako tomu bylo v předchozím experimentálním měření.



Graf 9: Změna průtoku při změně vstupního napětí na čerpadle



Graf 10: Změna průtoku ovlivňující teplotu vody v nádobě s topným tělesem

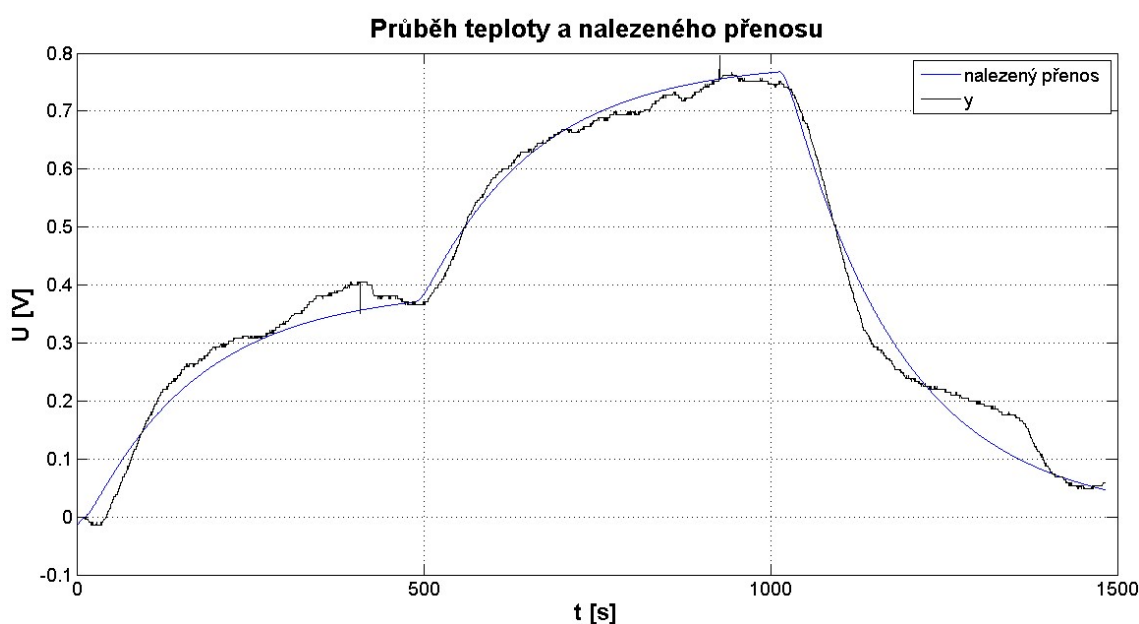


Graf 11: Průběh teploty při změně průtoku měřený teplotním čidlem (Arduino)

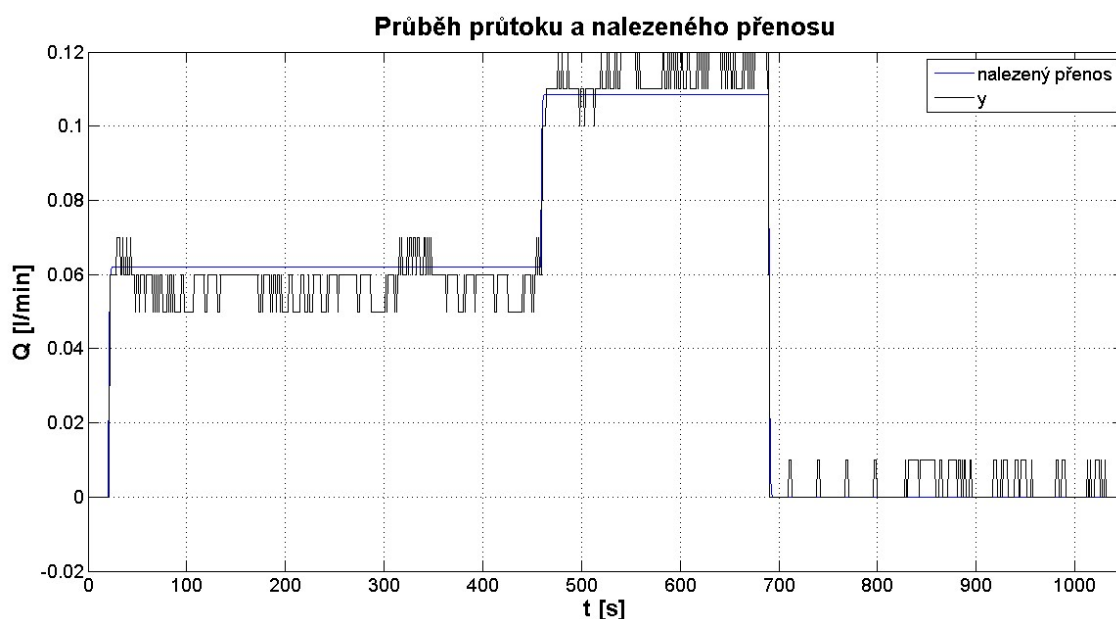
Z naměřených a zpracovaných dat se získaly příslušné grafy, které se využijí k identifikaci jednotlivých přenosů zkoumaného systému.

3.4 Identifikace systému

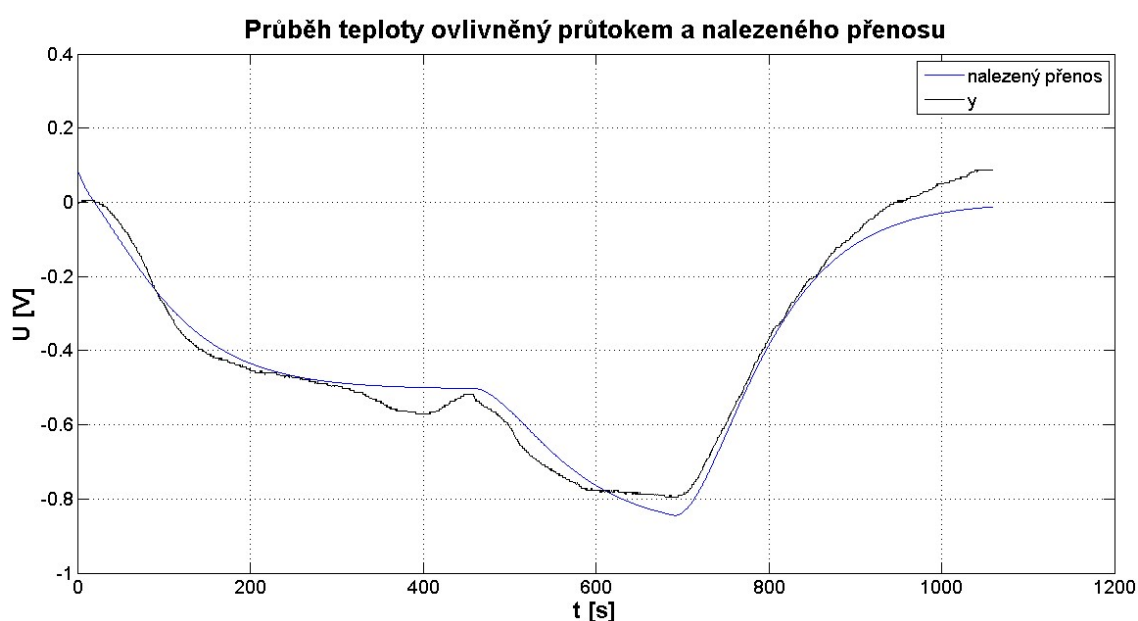
Pro identifikaci byl zvolen vhodný průběh skokových změn v lineární části okolo pracovních bodů, aby byla možná identifikace na základě reakcí na zvolené skokové změny. Popis a průběh měření je již popsán v předešlé kapitole, z které se bude vycházet při následujících identifikacích. K samotné identifikaci se využívá program Matlab a identifikační toolbox, takzvaný „Identification toolbox“. S jehož pomocí se stanoví jednotlivé přenosy systému. Do toolboxu se postupně naimportují naměřená data (t , u a y) a následně se pomocí vhodně zvolené estimace provede nalezení příslušných přenosových funkcí. Ve všech třech případech šlo o systém druhého řádu s reálnými kořeny. Následující grafy 12, 13 a 14 zobrazují porovnání naměřených dat s odezvou nalezeného přenosu. Černá barva je pro naměřená data a modrou barvou jsou označeny nalezené přenosy.



Graf 12: Porovnání shody naměřených hodnot teploty s nalezeným přenosem F_{22}



Graf 13: Porovnání shody naměřených hodnot průtoku s nalezeným přenosem F_{11}



Graf 14: Porovnání shody naměřených hodnot teploty ovlivněných průtokem s nalezeným přenosem F_{12}

Přenos F_{11} (graf 13) vyjadřuje závislost mezi napětím na čerpadle a velikostí snímaného průtoku. První křížovou vazbu představuje přenos F_{12} (graf 14). Jedná se o závislost mezi napětím na čerpadle, respektive rychlosti čerpání, a snímané teploty kapaliny

v nádobě s topnou spirálou. Přenos F_{21} představuje druhou křížovou vazbu, která je rovna nule, jelikož změnou teploty nedochází k ovlivnění velikosti průtoku. Poslední přenos F_{22} (graf 12) představuje závislost mezi napětím na napěťově řízeném zdroji přivedeného na topné těleso a snímanou teplotou kapaliny. Takto získané přenosy jsou zapsány do matice, která je ve tvaru rovnice 8. Statická zesílení stanovená z přenosové matice jsou v maticovém tvaru v rovnici 9.

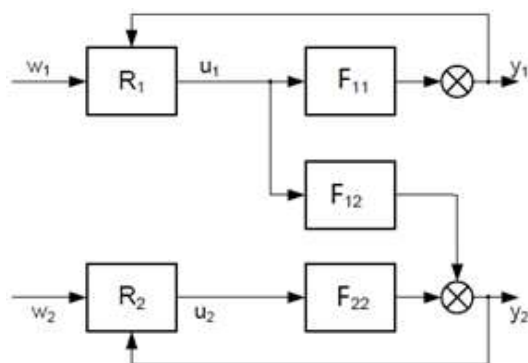
$$F = \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} = \begin{bmatrix} \frac{0,3}{(0,8 * s + 1) * (0,8 * s + 1)} & -\frac{2,5}{(76 * s + 1) * (21,8 * s + 1)} \\ 0 & \frac{0,7}{(103,3 * s + 1) * (28,8 * s + 1)} \end{bmatrix} \quad (8)$$

$$F(0) = \begin{bmatrix} 0,3 & -2,5 \\ 0 & 0,7 \end{bmatrix} \quad (9)$$

3.5 Decentralizované řízení

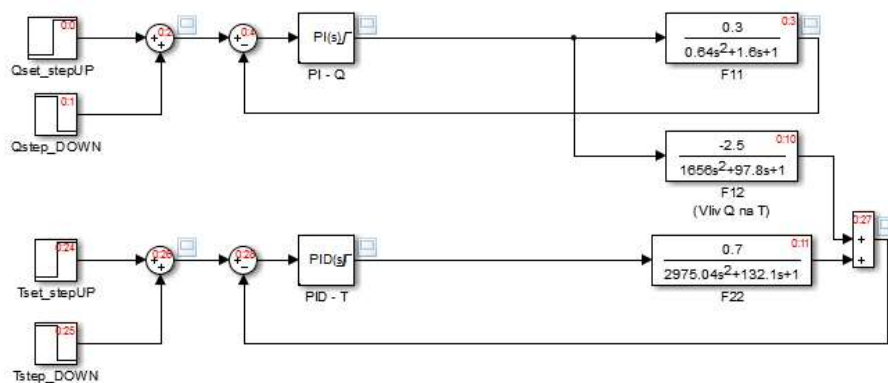
3.5.1 Jednoduché SISO smyčky

Základem decentralizovaného řízení je využití klasické PID regulace. Jedná se tedy o snahu získat z více rozměrového řízení systém, který je tvořen dvojicí nezávislých SISO smyček. Jedná se o jeden z nejjednodušších způsobů, jak řídit MIMO systém, který je složen ze dvou regulátorů. Každý z obou regulátorů je pak zapojený v samostatné smyčce. S tím souvisí volba přiřazení jednotlivých výstupů (y_1 a y_2) k příslušným vstupům (u_1 a u_2). Výsledné vazby jednotlivých SISO smyček jsou již předem jasné. Tedy k akčnímu vstupu u_1 se přiřadí výstup y_1 a k akčnímu vstupu u_2 se přiřadí výstup y_2 . Výsledkem je blokové schéma, které je na obrázku 19.



Obrázek 19: Blokové schéma SISO smyček s křížovou vazbou

Nastavení regulátoru R_1 vychází z nastavení regulátoru pro samostatnou regulaci průtoku, která je řešena v předešlé kapitole. Po té se provedlo nastavení druhého regulátoru R_2 . Nastavení parametrů obou regulátorů vychází z modelu, který je realizovaný v programu Matlab & Simulink. Zrealizovaný model je vidět na obrázku 20. Na základě modelu a jeho chování se provedlo nastavení obou regulátorů. Současně bylo snahou při nastavování obou regulátorů potlačení vlivu křížové vazby (F_{12} vliv Q na T), jak je vidět z obrázku 20.

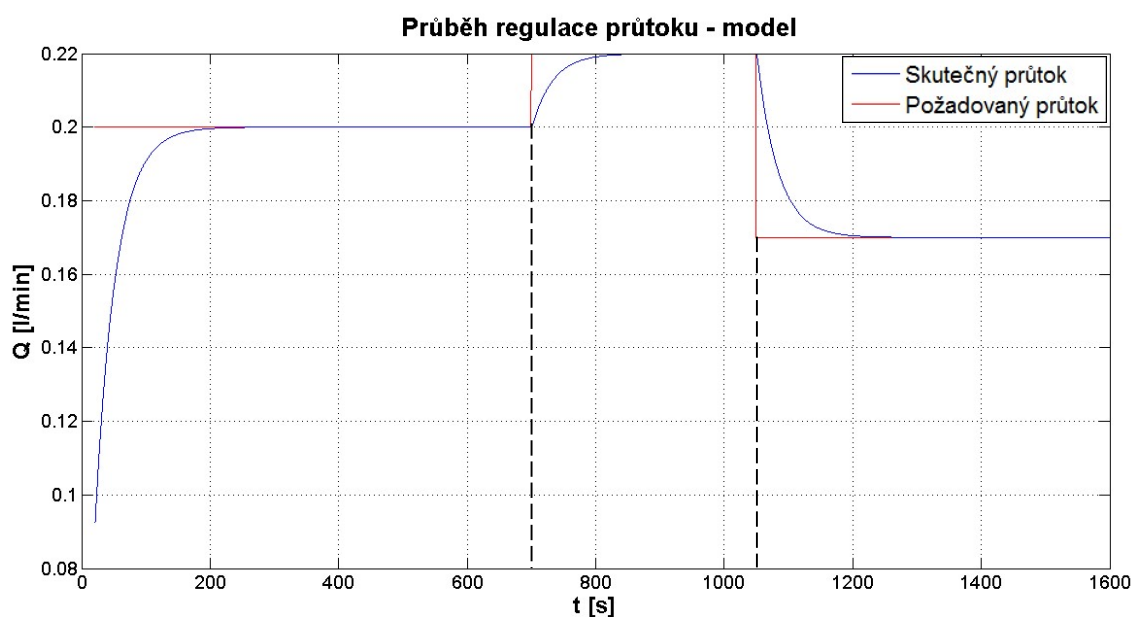


Obrázek 20: Schéma modelu v programu Matlab & Simulink

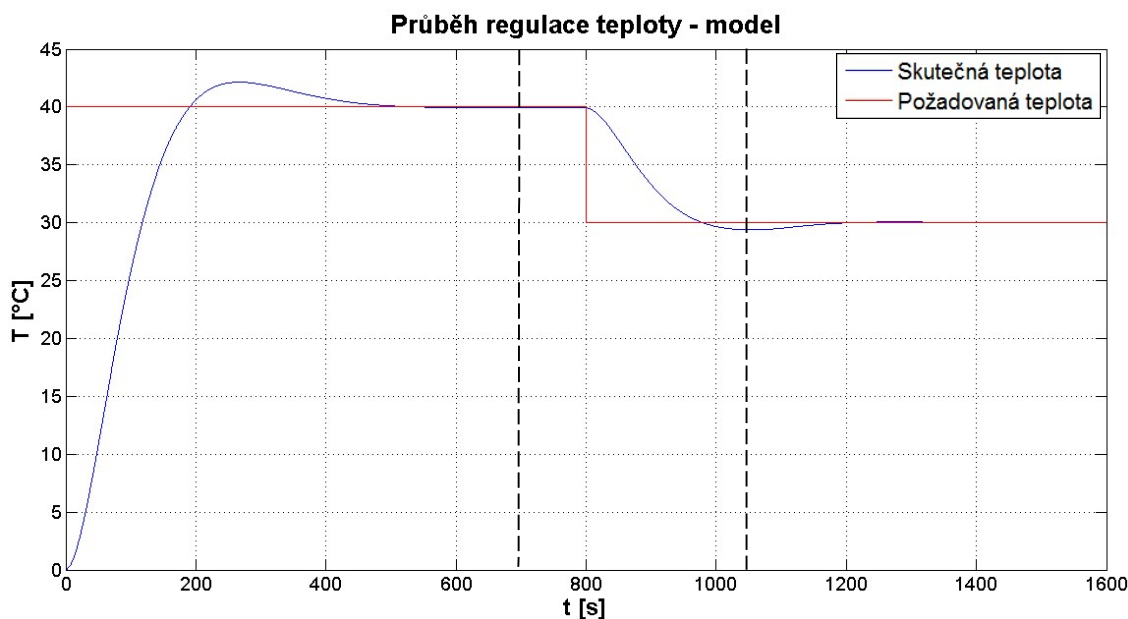
K regulaci průtoku se využívá PI regulátor, jehož parametry jsou $P_Q = 0,05$ a $I_Q = 0,10$. Parametry se získali pomocí ručního nastavení ve spojení s autotuningem v programu Matlab & Simulink. Výsledný průběh modelového regulátoru průtoku je vidět v grafu 15. Červenou barvou je označena žádaná hodnota a modrou barvou je vyznačen průběh

skutečné hodnoty. Toto barevné značení je stejné i pro následující grafy s příslušnými průběhy.

Následně se provedlo nastavení regulátoru teploty, který je navíc ještě ovlivněn negativním působením křížové vazby. K regulaci teploty vody v nádobě s topným tělesem je možné použít PI regulátoru s parametry $P_T = 1,60$ a $I_T = 0,02$. Avšak mnohem vhodnější je využití PID regulátoru s parametry $P_T = 1,60$, $I_T = 0,02$ a $D_T = 1,40$. Parametry se získaly stejným způsobem, jako tomu bylo u regulace průtoku. Výsledný průběh regulace teploty spolu s nežádoucím působením křížové vazby je vidět z grafu 16. Výsledný negativní vliv průtoku na teplotu není příliš velký, dle modelového průběhu, ale i tak je patrný a negativně působí při regulaci teploty. Z grafu 16 je vidět, že negativní vliv se projevil v oblasti časové osy v čase $t = 700$ s a dále pak je vidět negativní vliv okolo času $t = 1100$ s. Z grafů je dále vidět již zmíněný fakt, že zde druhá křížová vazba není, tedy změna teploty neovlivňuje změnu průtoku.



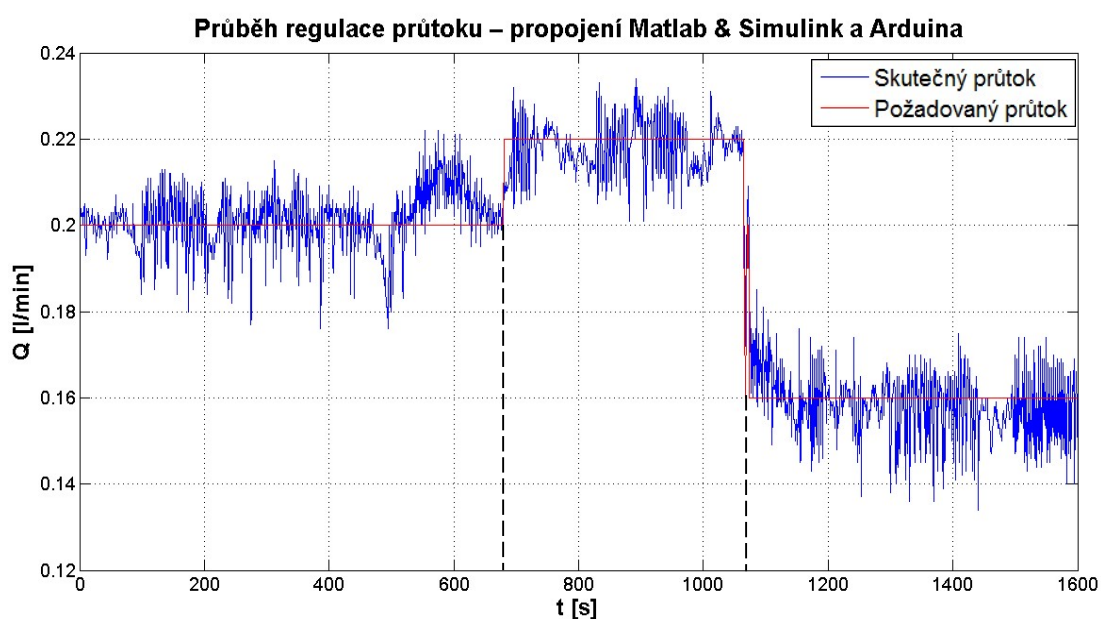
Graf 15: Regulace průtoku pomocí PI regulátoru



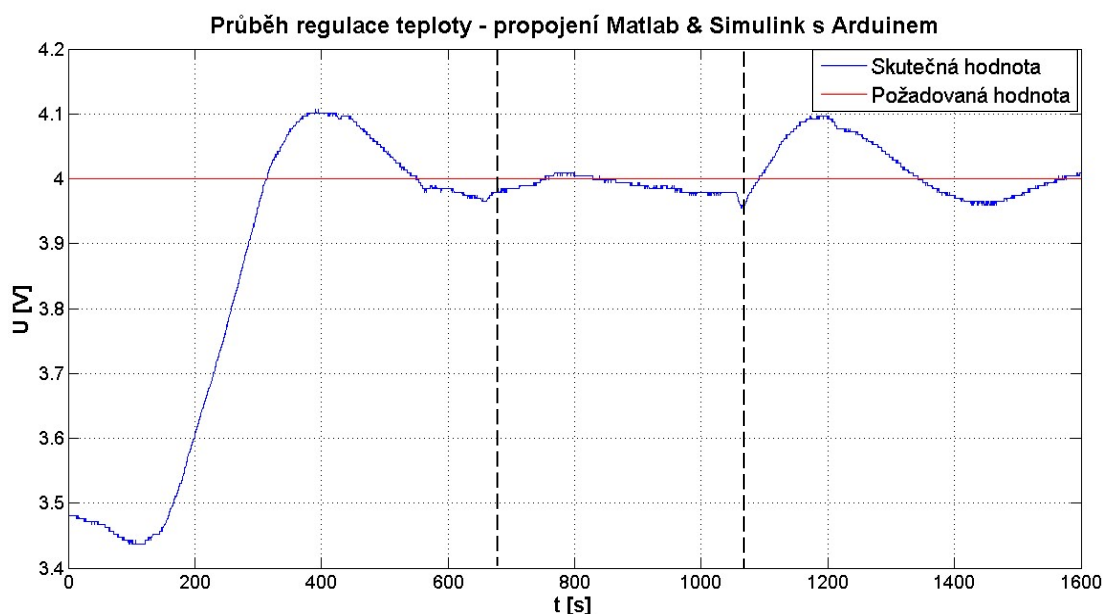
Graf 16: Regulace teploty PID regulátorem + vliv křížové vazby

Nyní je třeba celý proces zrealizovat pomocí Arduina propojeného s programem Matlab & Simulink a následně provést realizaci celého procesu pouze s využitím Arduina. Nejprve se propojí model s Arduinem pomocí měřicí karty PCI 1711, což umožní regulovat teplotu pomocí modelu bez využití regulátoru zrealizovaného pomocí Arduina, zatímco regulace průtoku se provádí regulátorem, který je naprogramovaný v Arduinu. Parametry regulátoru PI a PID jsou stejné, jako tomu bylo u modelového nastavení obou regulátorů. Výsledné průběhy regulací jsou pak vidět z grafů 17 a 18. Graf 17 zobrazuje průběh regulace průtoku pomocí Arduina. Regulace průtoku je bohužel ovlivněna ne zcela vhodně nastaveným PI regulátorem průtoku a dále pak průtokoměrem samotným, jehož nestandardní chování se negativně projevilo i při dalším měření. Současně je zde i snaha otestovat regulátor průtoku i na poruchu, tedy odolnost vůči vnějším poruchám. Proto při experimentálním měření není nádoba s čerpadlem zcela naplněna vodou. Dojde-li k rapidnímu poklesu výšky hladiny, dojde i k poklesu průtoku během regulace, což je vidět z grafu 17 v čase $t = 450$ s. Naopak při velmi rychlém napuštění vody náraz do nádoby s čerpadlem dojde k rapidnímu nárůstu výšky hladiny a tím i nárůstu průtoku. Tento jev je opět vidět z grafu 17 v čase $t = 500$ s. Je vidět, že vyvolané změny se projeví ve změně průtoku o $\pm 0,03$ l/min. Za normálních podmínek, kdy je nádoba po celou dobu provozu ve stejné výšce a voda se odčerpává jen za pomoci samotného čerpadla, je regulátor schopen potlačit poruchu způsobenou poklesem hladiny. To samé

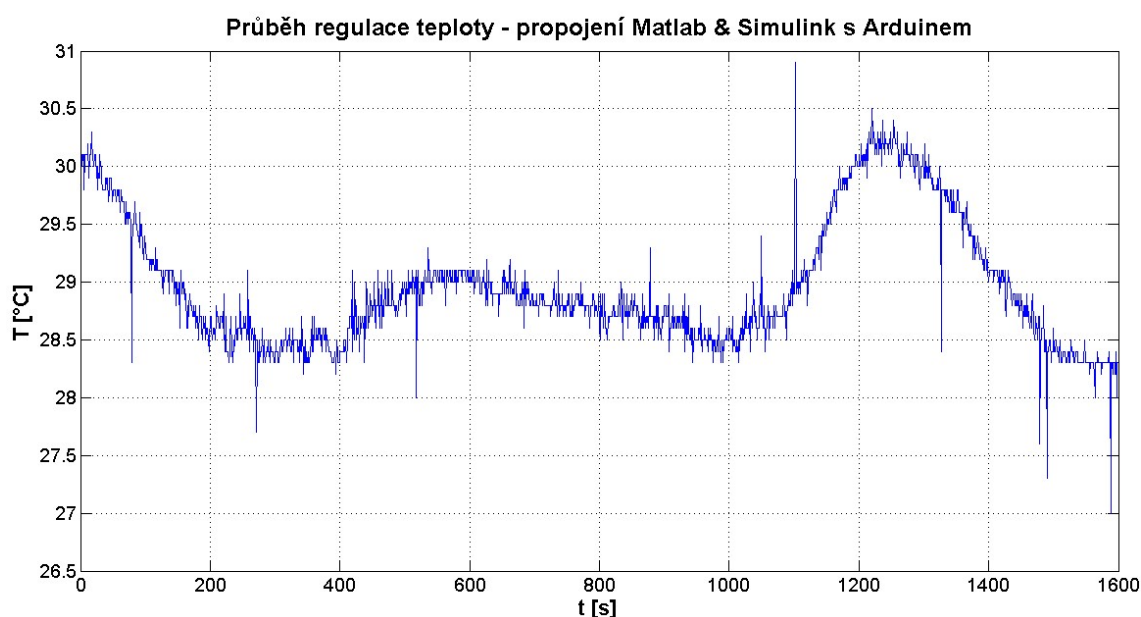
platí i o dopouštění vody do nádoby, kdy je voda dopouštěna postupně a nikoli naráz. Z grafu 18 je pak vidět zaznamenaný průběh regulace teploty za pomoci modelového PID regulátoru a teplotního čidla PT 100. Zde je již mnohem více patrný vliv křížové vazby. Při skokové změně průtoku dojde k poklesu teploty v čase $t = 700$ s. Mnohem znatelnější vliv křížové vazby je při změně průtoku o 0,05 l/min v čase $t = 1050$ s, kdy došlo k překmitu o zhruba $+ 1,5$ °C oproti žádané hodnotě teploty. Průběh teploty zaznamenaný ve stupních Celsia pomocí Arduina a teplotního čidla je pak zaznamenan do grafu 19.



Graf 17: Průběh regulace průtoku



Graf 18: Regulace teploty za využití modelu spojeného s platformou

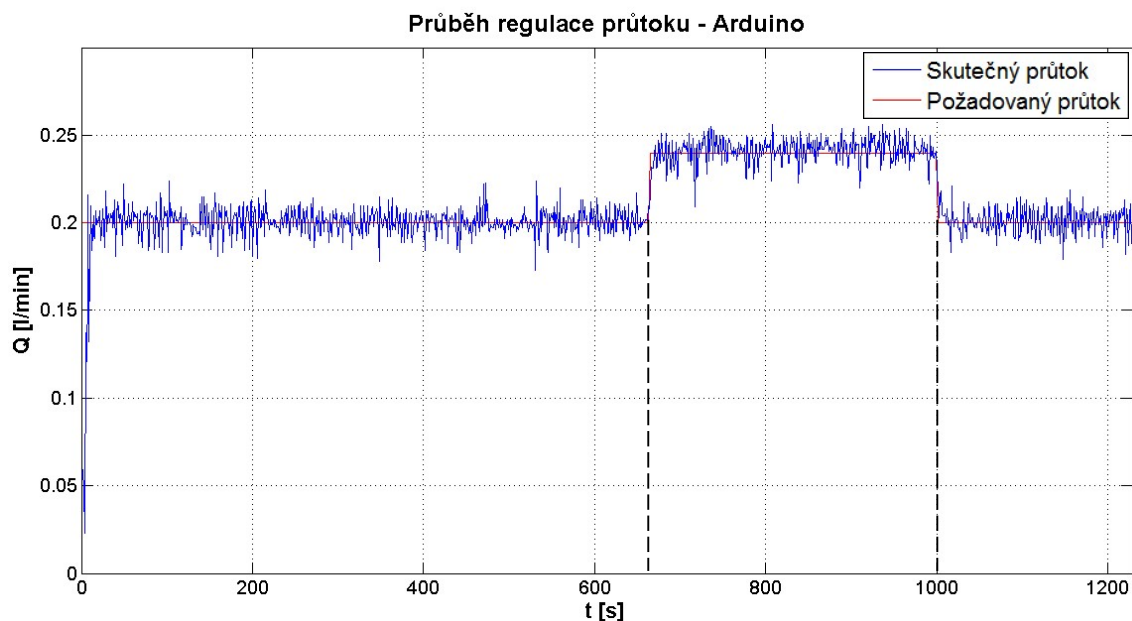


Graf 19: Průběh regulované teploty měřené pomocí Arduina a teplotního čidla

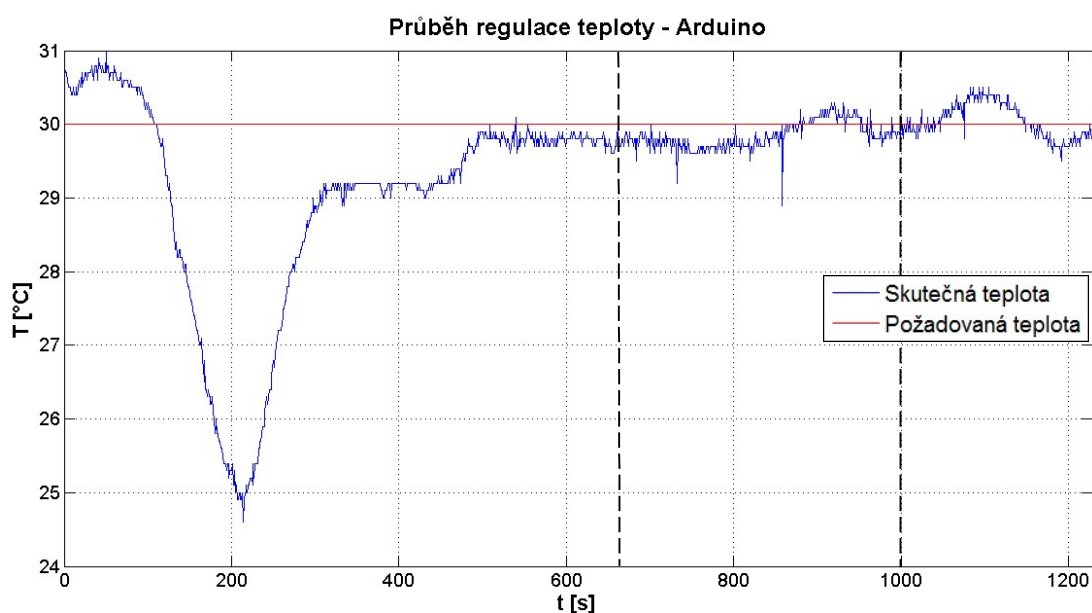
Nyní ještě zbývá provést zrealizování regulace teploty a průtoku pouze za pomoci Arduina, již bez využití měřící karty PCI 1711 a modelu. Parametry pro regulátor teploty a průtoku jsou nastaveny stejně, jako tomu bylo v předchozích experimentálních měření a testech s modelem. Data získaná z tohoto experimentálního měření byla zpracována a vynesena do několika grafů. Z grafu 20 je možné vidět průběh regulace průtoku. Opět

červenou barvou je označena žádaná hodnota, která je v úvodu nastavena na hodnotu $Q = 0,20$ l/min. Následně je provedena skoková změna na hodnotu průtoku 0,24 l/min. Modrou barvou je pak označena skutečná hodnota průtoku. Z grafu je vidět relativně rychlá reakce regulátoru na provedenou změnu průtoku a dále je vidět, že graf není nijak zásadně ovlivněn změnou výšky hladiny v nádobě s čerpadlem. Zvolené hodnoty P_Q a I_Q se ukázaly být vhodnými pro tuto úlohu. Graf 21 představuje průběh regulace teploty, který je současně ovlivněn negativním působením křížové vazby. Červenou barvou je opět označena žádaná hodnota teploty, která je nastavena na 30 °C a opět modrou barvou je označen skutečný průběh výstupní teploty, měřený teplotním čidlem z Arduina. Výstupní průběh teploty regulovaný PID regulátorem z grafu 21 je ovlivněn již zmíněnou křížovou vazbou. Nejprve se projeví její vliv kolem času $t = 650$ s, kdy s rostoucím průtokem dochází k poklesu teploty. V čase $t = 1000$ s se projeví vliv průtoku na teplotu podruhé, ovšem mnohem zásadněji, než jak tomu bylo v prvním případě, kdy vliv křížové vazby nebyl tolik patrný. I zde se ukázalo, že volba parametrů P_T , I_T a D_T byla správná.

I přes snahu vhodného nastavení jednotlivých regulátorů nebylo dosaženo potlačení vlivu křížové vazby, proto je třeba využít a otestovat některou z metod vedoucí k potlačení a eliminaci křížové vazby.



Graf 20: Regulace průtoku pomocí Arduino

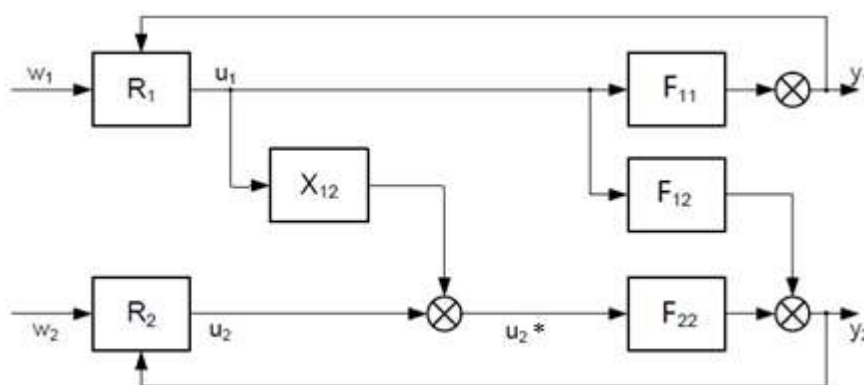


Graf 21: Regulace teploty pomocí Arduino

3.5.2 Dekompozice

Dekompoziční metoda se využívá pro zpětnovazební MIMO systém. Při detailnějším pohledu na blokové schéma z obrázku 19 či 20 je patrné, že se zde nejedná o zcela typický MIMO systém, jelikož zde vypadla jedna křížová vazba. I přesto se dá na celý systém nahlížet jako na MIMO systém, který má dva vstupy a dva výstupy a pouze jed-

nu křížovou vazbu. Navíc pro tento případ je dekompozice shodná s realizací regulátoru pro odstranění měřené poruchy. Systém je opět řízen pomocí dvou regulátorů. Stejně tak, jako tomu bylo v předchozí metodě pro jednoduché SISO smyčky. Jen s tím rozdílem, že cílem této metody je najít matici filtru pro stávající systém. Úkolem filtru pak bude potlačit a eliminovat negativní vliv křížové vazby. Výsledné blokové schéma včetně filtru X_{12} je na obrázku 21. Do filtru X_{12} vstupuje akční veličina u_1 a výstup z filtru X_{12} je přičten k akční veličině u_2 , čímž generuje akční veličinu u_2^* a tak dochází k potlačení a eliminaci nežádoucích účinků křížové vazby F_{12} .



Obrázek 21: Blokové schéma celého systému s filtrem X_{12}

Nejprve je nutné stanovit výsledný přenos filtru X_{12} , který vychází ze stanoveného přenosu z rovnice 10.

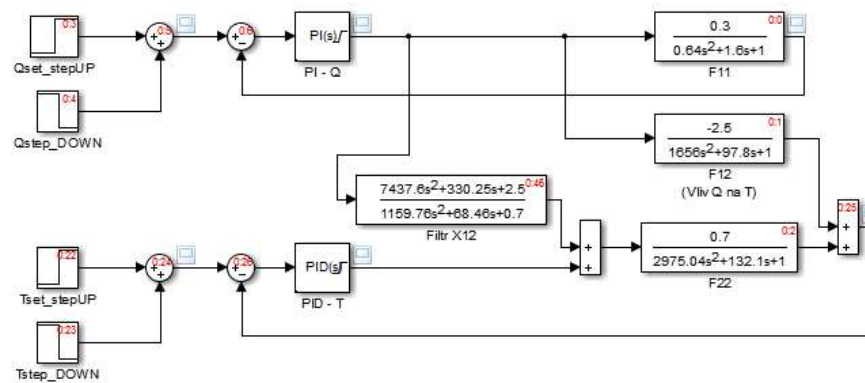
$$F_{w_1 y_2} = \frac{F_{12} + X_{12} * F_{22}}{1 + R_2 * F_{22}} \quad (10)$$

Dále se rovnice 10 upraví do tvaru, jehož výsledkem bude přenos filtru. Parametry jmenovatele nelze ovlivnit a požadavek na přenos $F_{w_1 y_1}$ je takový, aby byl roven nule, čímž se získá rovnice 11. Z této rovnice se následně získá výsledný přenos filtru X_{12} , který je vidět z rovnice 12.

$$0 = F_{12} + X_{12} * F_{22} \quad (11)$$

$$X_{12} = -\frac{F_{12}}{F_{22}} = \frac{7437,60 * s^2 + 330,25 * s - 2,5}{1159,76 * s^2 + 68,46 * s + 0,70} \quad (12)$$

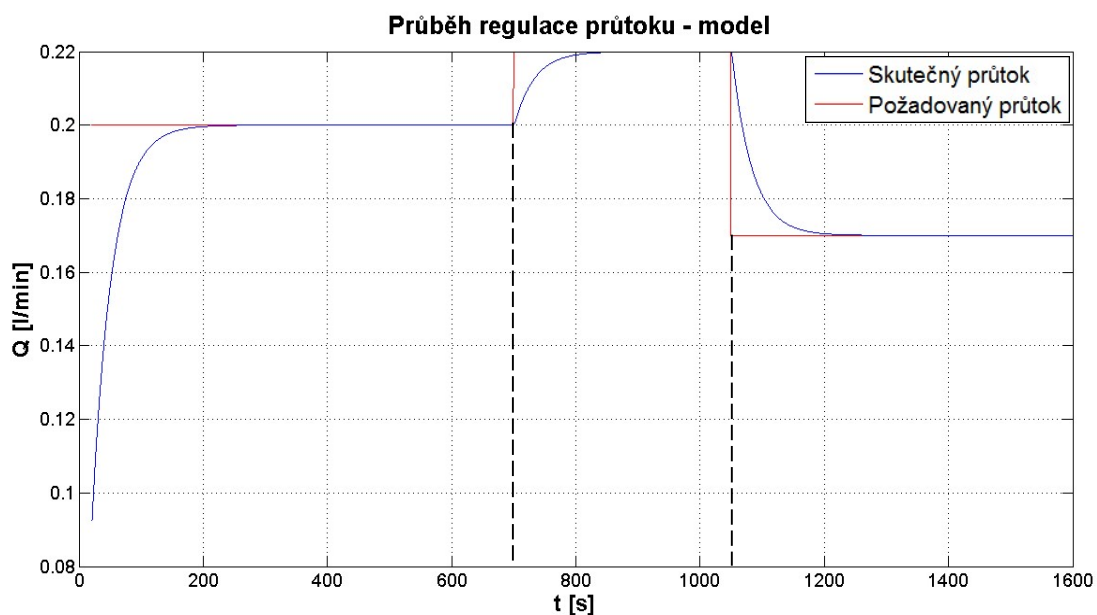
Nyní se může provést prvotní testování funkce filtru za pomoci modelu realizovaného v programu Matlab & Simulink. Zrealizovaný model včetně filtru je na obrázku 22.



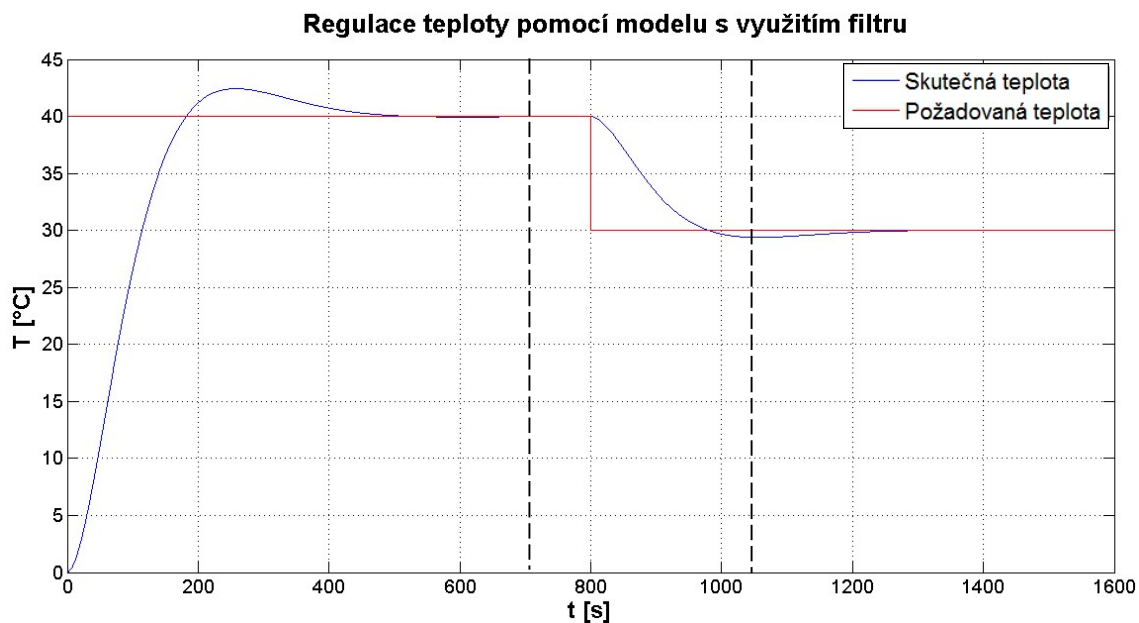
Obrázek 22: Schéma modelu včetně filtru v programu Matlab & Simulink

Parametry pro regulátor průtoku a teploty jsou stejné, jako tomu bylo u předešlých experimentálních měření. Červeně jsou opět označeny žádané hodnoty a modře jsou označeny skutečné hodnoty průtoku a teploty. Z modelového průběhu regulace průtoku, který je zaznamenán a vyneseno do grafu 22, je vidět, že zde nenastala žádná změna, což není nic nečekaného. Naopak změna nastala v průběhu modelové regulace teploty, jak je možné vidět z grafu 23. Filtr zde zcela potlačil a eliminoval negativní vliv křížové vazby. Nyní již nedochází při změně průtoku k ovlivňování regulace teploty, jak tomu bylo při modelové regulaci bez využití filtru, viz graf 16.

Nyní je prokázána funkčnost filtru na modelové situaci, proto je třeba provést opět testování nejprve s využitím Arduina propojeného s modelem za pomoci měřicí karty PCI 1711 a ověřit funkčnost celku. Posledním krokem bude vhodná programová implementace filtru do mikroprocesoru a následné otestování, kdy celý proces regulace včetně filtrace bude prováděn čistě pomocí Arduina.



Graf 22: Průběh regulace průtoku – model



Graf 23: Průběh regulace teploty opatřený filtrem

Aby bylo možné filtr implementovat do Arduina, je třeba jej převést do diskrétního tvaru rovnice 14. K tomuto tvaru rovnice se dostalo následujícím postupem. Nejprve je třeba převést rovnici 12, která je ve tvaru spojité funkce. Pomocí programu Matlab a funkce `c2d()` se převede spojitý tvar rovnice na diskrétní tvar, který je ještě upraven

do tvaru rovnice 13. Z takto získané rovnice je možné vhodným přepisem získat výsledný tvar diskrétní rovnice 14.

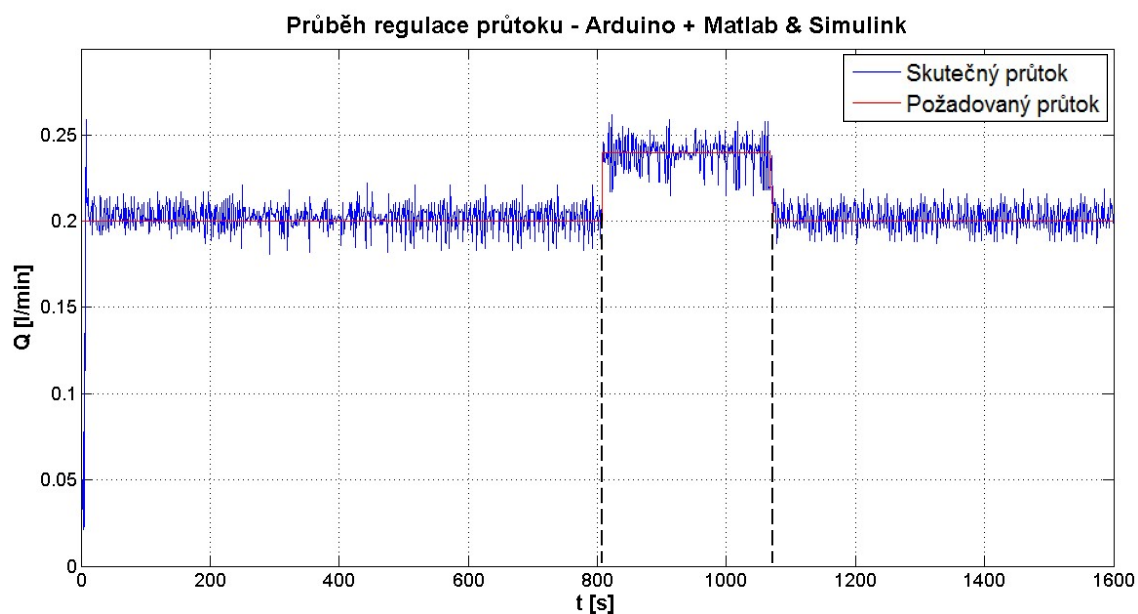
$$G(z) = \frac{6,413 * z^0 - 12,69 * z^{-1} + 6,272 * z^{-2}}{z^0 - 1,971 * z^{-1} + 0,9709 * z^{-2}} \quad (13)$$

$$\begin{aligned} y[n] = & 1,971 * y[n-1] - 0,9709 * y[n-2] + 6,413 * x[n] \\ & - 12,69 * x[n-1] + 6,272 * x[n-2] \end{aligned} \quad (14)$$

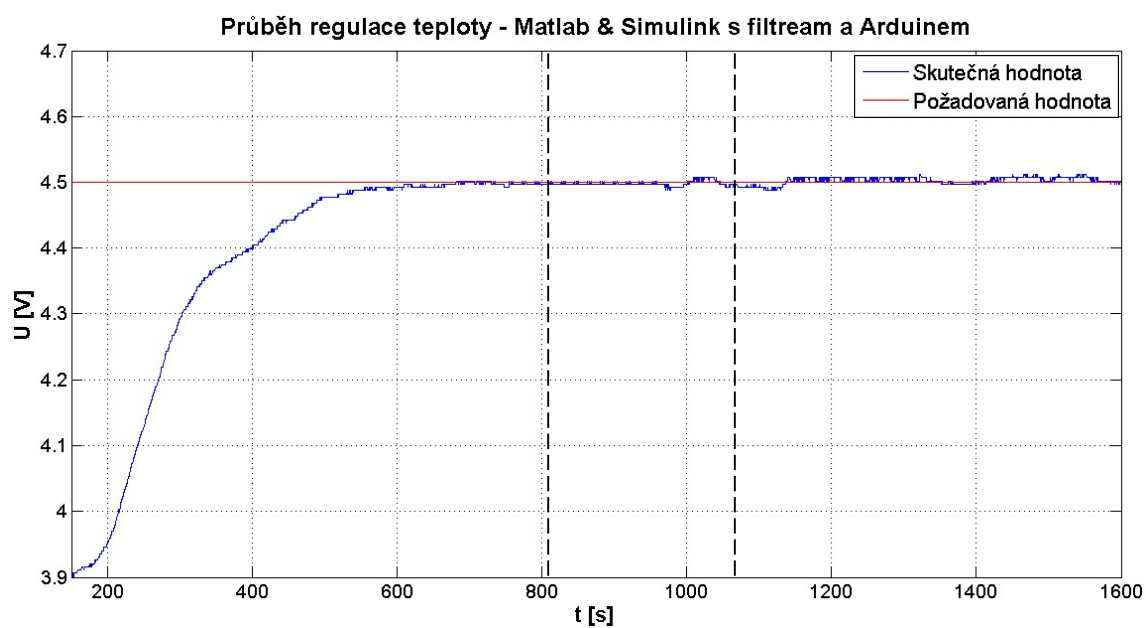
V takovémto tvaru se dá celá rovnice mnohem lépe zrealizovat v programové části a následně nahrát na mikrokontrolér. Výsledný filtr je realizován ve formě knihovny, která se naimplementuje do celého programu a zároveň se do smyčky *Setup* vloží několik řádků kódu, které spustí filtr a nadefinují potřebné proměnné. Pomocí funkce *Filtr* se zavolá část programu, která slouží k potlačení a eliminaci křížové vazby. Do filtru pak vstupuje výstupní hodnota z programově řešeného PI regulátoru průtoku. Po zpracování signálu filtrem, se výstupní hodnota z filtru přičte k výstupu z programově řešeného regulátoru průtoku. Takto nově vzniklá výstupní hodnota se запиše na příslušný výstupní Arduino pin10, kam je připojen vstup napětově řízeného zdroje napětí.

Nyní je třeba přejít k dalšímu mezikroku, kdy se model v programu Matlab & Simulink propojí měřicí kartou PCI 1711 se systémem, díky čemuž se využijí modelové regulátory namísto těch realizovaných pomocí Arduina. Teplota je snímána pomocí teplotního čidla PT100, které je přivedeno na měřicí kartu, stejně tak, jako tomu bylo již v předchozích experimentálních měřeních. Výsledkem měření jsou tři průběhy, které jsou vyneseny do příslušných grafů. Z grafu 24 je vidět průběh regulace průtoku, kdy je žádaná hodnota průtoku nejprve nastavena na hodnotu 0,20 l/min. Následně je provedena skoková změna žádané hodnoty průtoku na hodnotu 0,24 l/min a následuje skoková změna žádané hodnoty průtoku zpět na původní hodnotu 0,20 l/min. Získaný průběh regulace průtoku opět nepřináší nic neočekávaného, jelikož zde nedošlo k žádným změnám. I zde platí, že parametry obou regulátorů jsou ponechány tak, jak tomu bylo u modelu. Z grafu 25, který zobrazuje regulaci teploty s využitím filtru, je vidět, že filtr splnil požadavky v tomto mezikroku a dokázal potlačit negativní vliv křížové vazby, tedy negativní vliv změny průtoku na teplotu. Nedochází tak k ovlivnění teploty při změně prů-

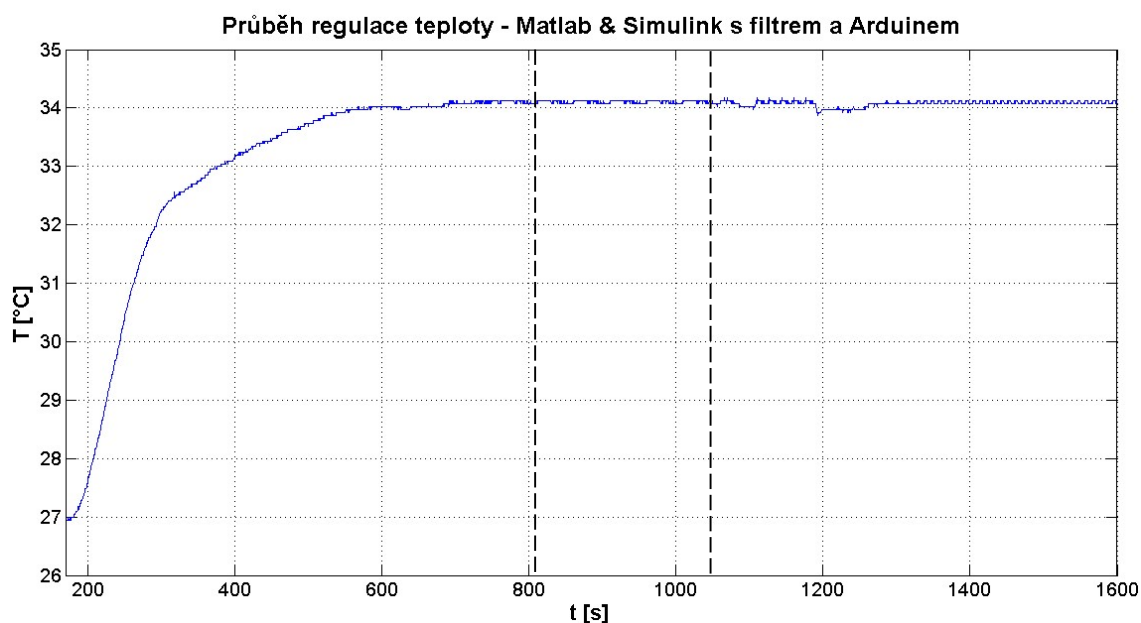
toku, čímž je prokázána funkčnost filtru s využitím modelu a měřicí karty PCI 1711. Součástí experimentálního měření je i graf 26, na kterém je průběh teploty ve stupních Celsia, který je měřený pomocí teplotního čidla a Arduina. Měřená teplota se opět shoduje s průběhem z grafu 25. Jelikož je prokázána funkce filtru v kombinaci modelu s měřicí kartou, může se přejít k testování za pomoci samotného Arduina, na kterém jsou realizovány oba regulátory včetně filtru.



Graf 24: Regulace průtoku pomocí PI regulátoru a Arduina

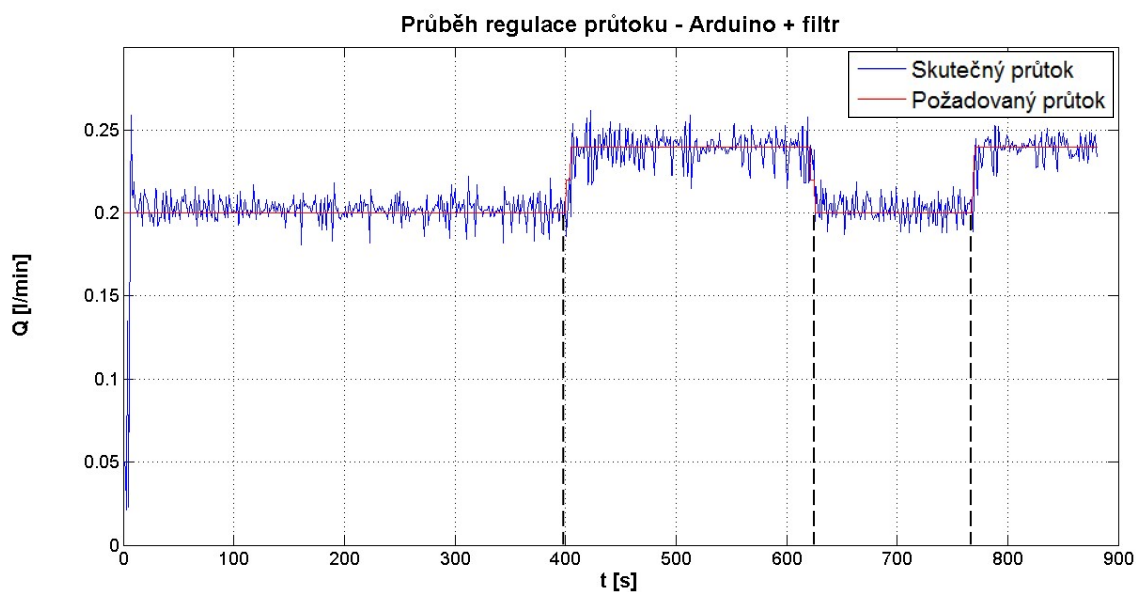


Graf 25: Regulace teploty pomocí PID modelového regulátoru + filtr

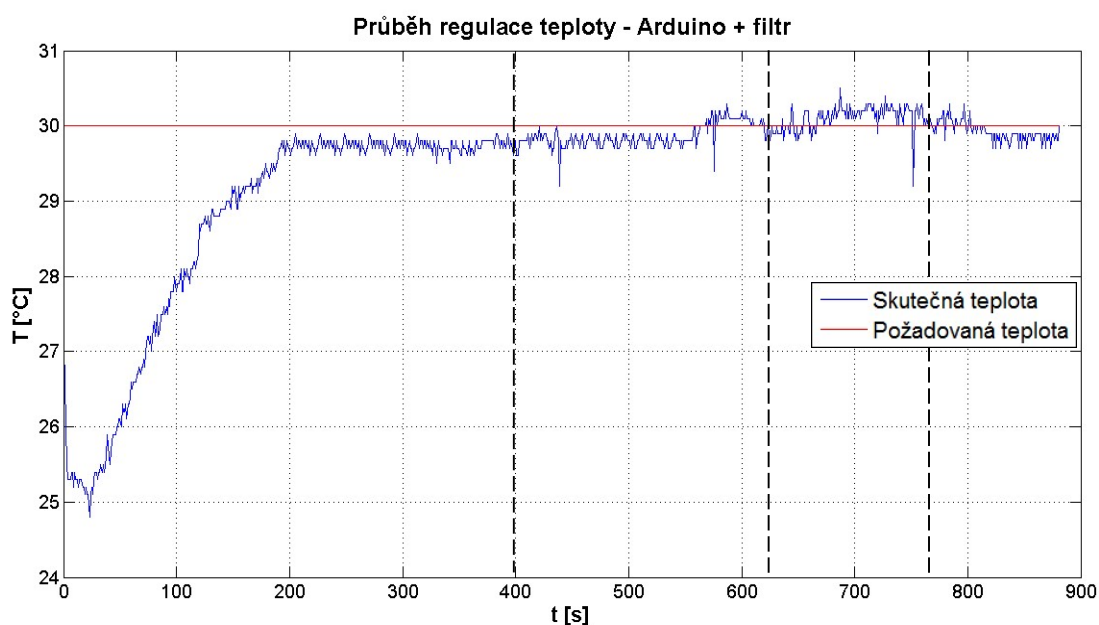


Graf 26: Průběh teploty naměřený pomocí Arduino teplotního čidla

Posledním krokem je ověření celého experimentu regulace s využitím filtru pouze za pomoci Arduina. Výsledkem jsou dva grafy. Graf 27 představuje průběh regulace průtoku pomocí PI regulátoru. Parametry P_Q a I_Q jsou stejné, jako tomu bylo u předešlých nastaveních regulátoru průtoku. Výsledný průběh regulace průtoku je více méně stejný, jako tomu bylo u předešlých experimentálních měření. Regulace průtoku není nijak ovlivněna a rovněž není nijak znát změna výšky hladiny v nádobě s čerpadlem. Z druhého grafu tohoto experimentu je vidět průběh regulace teploty pomocí PID regulátoru. I zde byly parametry P_T , I_T a D_T ponechány beze změny. Výsledný průběh je pak vyneseno do grafu 28. Z grafu je vidět, že filtr dokázal více či méně potlačit vliv křížové vazby, čímž nedochází k ovlivnění teploty při změně průtoku, jako tomu bylo v předchozí kapitole, kde byla prováděna regulace bez využití filtru.



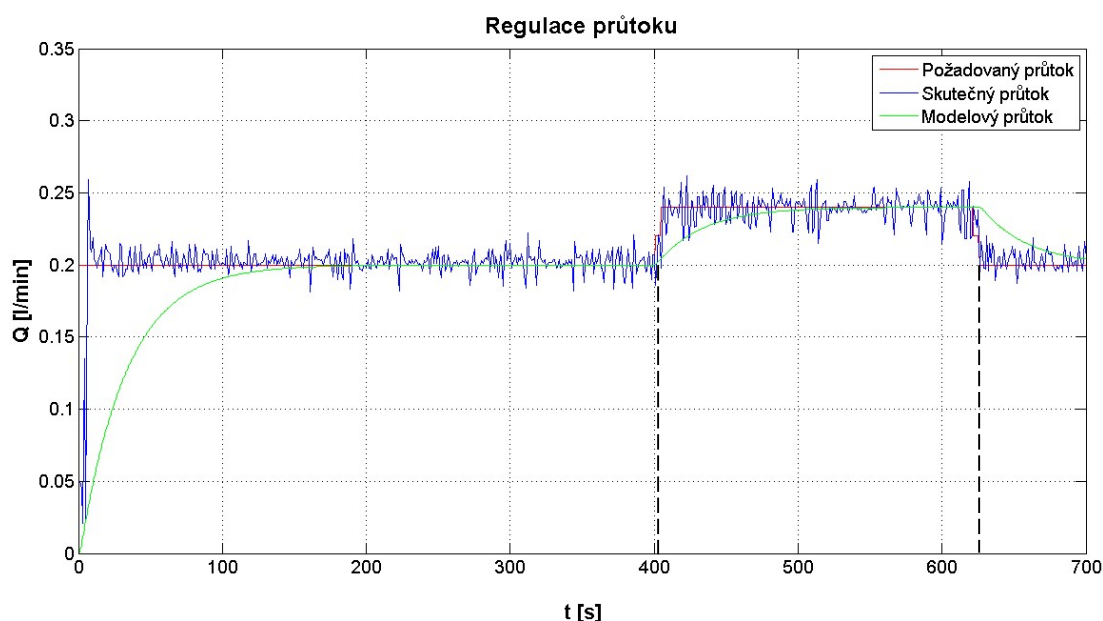
Graf 27: Regulace průtoku pomocí Arduina



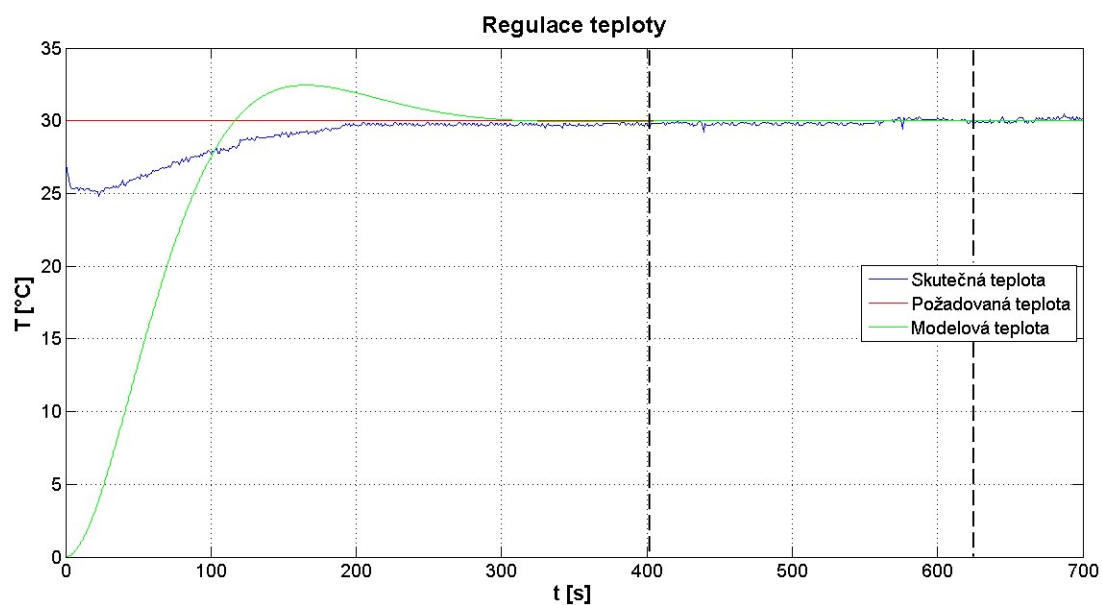
Graf 28: Regulace teploty pomocí Arduina s filtrem

Regulace průtoku a teploty bez využití filtru by mohla být do jisté míry postačující pro tuto úlohu, avšak i přesto zde byl patrný negativní vliv křížové vazby. Zařazením filtru došlo k potlačení negativního vlivu a tím i ke zlepšení regulace teploty, což je možné vidět, pokud porovnáme grafy 21 a 28. Z toho pak plyne jasný závěr, že využití filtru jasně vede ke zlepšení a je mnohem vhodnější při potlačení negativních vlivů křížové vazby.

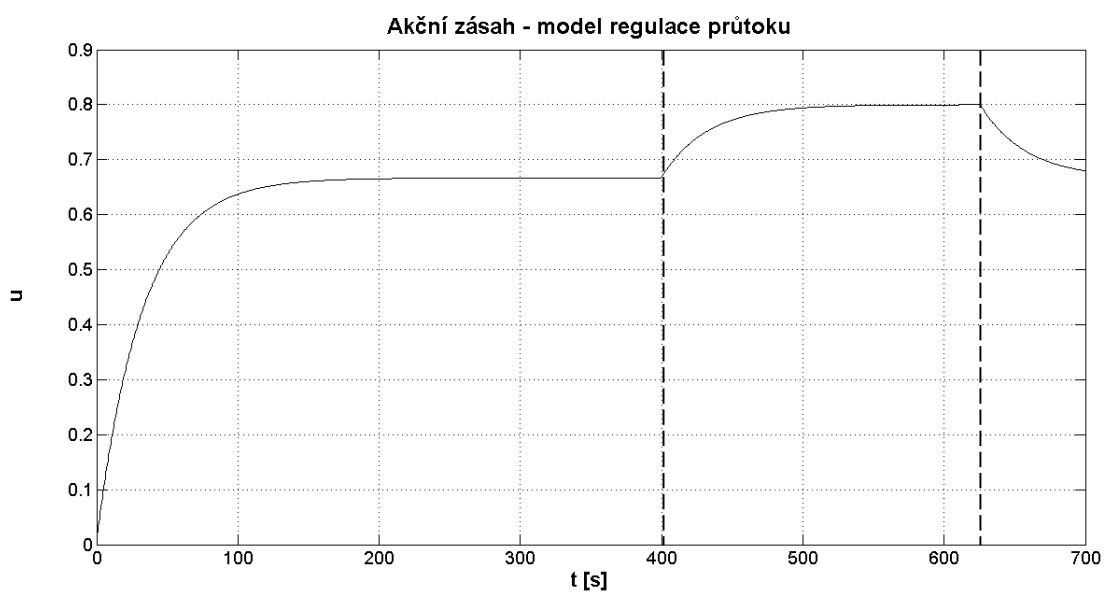
Ná úplný závěr této kapitoly je ještě dobré ukázat několik grafů, které slouží k porovnání modelových hodnot se skutečnými hodnotami, které se získaly při regulaci za pomoci samotného Arduina. Graf 29 porovnává modelový průběh regulace průtoku se skutečným průběhem. Z grafu je vidět, že skutečný průběh průtoku je více kmitavý nežli modelový průběh. Ovšem oproti modelové regulaci dokáže skutečný regulátor dosahovat mnohem rychleji požadovaných hodnot po provedení změny. To samé platí i pro následující graf 30, který porovnává opět modelový průběh se skutečným průběhem. Ovšem v tomto případě jde o průběhy regulace teploty. Reálný průběh regulace teploty je opět lehce kmitavý oproti modelu a opět mnohem rychleji dosahuje ustálených hodnot po změně oproti modelu. Poslední dvojice grafů 31 a 32 zobrazují průběh akčních zásahů regulátoru průtoku (graf 31) a průběh akčního zásahu regulátoru teploty (graf 32).



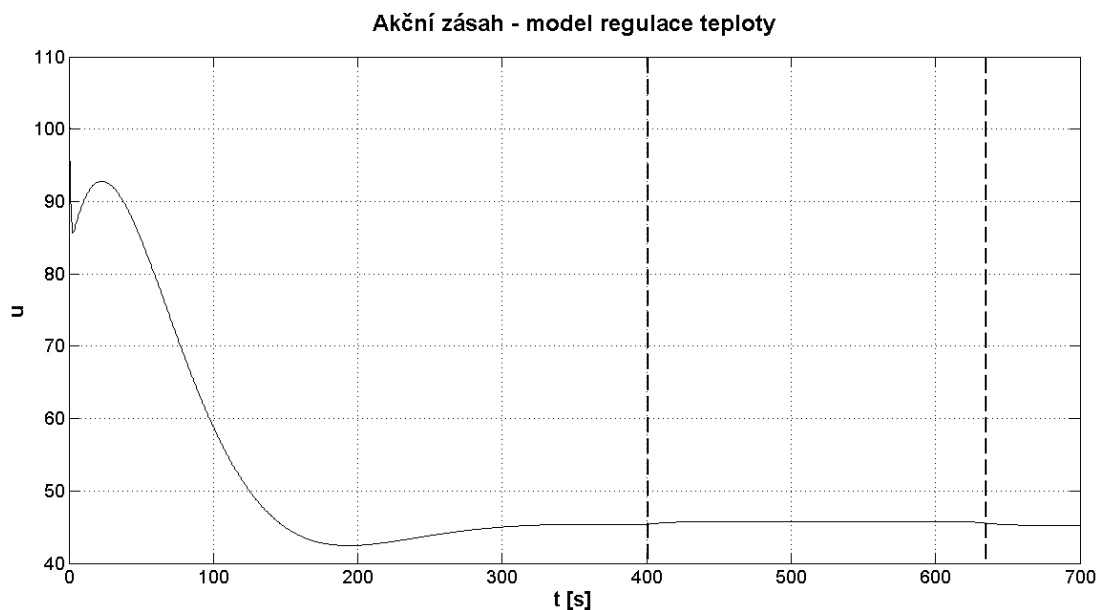
Graf 29: Porovnání modelových a reálných průběhů - regulace průtoku



Graf 30: Porovnání modelových a reálných průběhů - regulace teploty



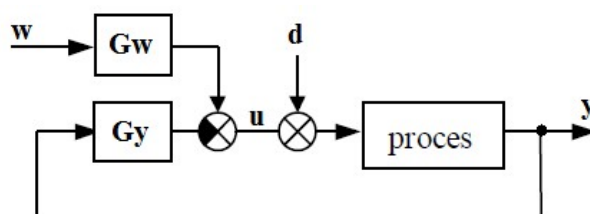
Graf 31: Průběh akčních zásahů - regulace průtoku



Graf 32: Průběh akčních zásahů - regulace teploty

3.5.3 Regulátor se dvěma stupni volnosti (2DOF regulátor)

Zapojení s využitím regulátoru s dvěma stupni volnosti je jeden z dalších možných způsobů, jak potlačit a eliminovat negativní vliv průtoku na teplotu (křížová vazba F_{12}). V tomto případě se bude na křížovou vazbu nahlížet jako na poruchu. Prvním požadavkem je potlačení poruchy a druhým požadavkem je správné sledování žádané hodnoty. Jedná se o odlišné požadavky, proto je pochopitelné, že není možné dosáhnout splnění obou požadavků současně s využitím zpětnovazebního regulátoru. Právě proto se k řešení tohoto problému využije regulátor s dvěma stupni volnosti, také označován jako 2DOF (Two Degrees Of Freedom) regulátor.



Obrázek 23: Principiální schéma regulátoru s dvěma stupni volnosti (2DOF)

Regulátor umožňuje zpracovávat oba vstupy (žádanou a regulovanou veličinu) nezávisle oproti klasickému zpětnovazebnímu regulátoru, který pracuje pouze s regulační odchylkou. Blokové schéma regulátoru s dvěma stupni volnosti je na obrázku 23.

Z obrázku je vidět, že změnou parametrů bloku s žádanou hodnotou nijak neovlivní stabilitu uzavřené smyčky. Současně nedochází ke změně tvaru odezvy na poruchu d . Jednou z možných variant PID regulátoru se dvěma stupni volnosti je ve tvaru rovnice 15. Odchylka pro výpočet proporcionální části se pak stanoví pomocí rovnice 16 a odchylka pro derivační složku se pak stanoví pomocí rovnice 17.

$$u(t) = K \left(e_p + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de_d}{dt} \right) \quad (15)$$

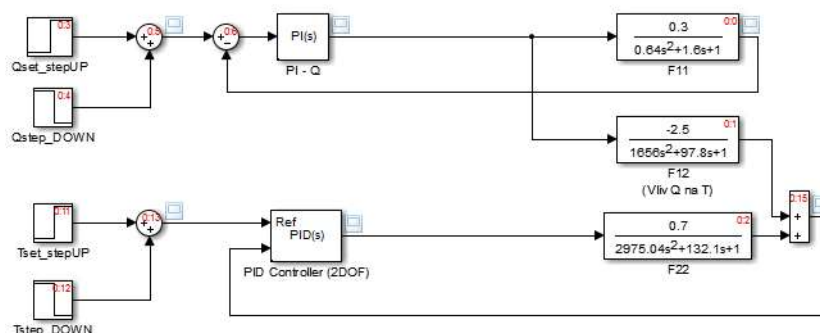
$$e_p = b * w - y \quad (16)$$

$$e_d = c * w - y \quad (17)$$

Algoritmus PID regulátoru tedy obsahuje pět parametrů K , T_i , T_d , b a c . První tři zmíněné parametry zajišťují optimalizaci potlačení poruchy a zbývající dva parametry slouží k tvarování odezvy na požadovanou hodnotu. Ještě je dobré zmínit, že u komerčně řešených PID regulátorů s dvěma stupni volnosti je většinou $c = 0$ a b je možné libovolně volit, nebo je omezeno pouze na 0, nebo 1.

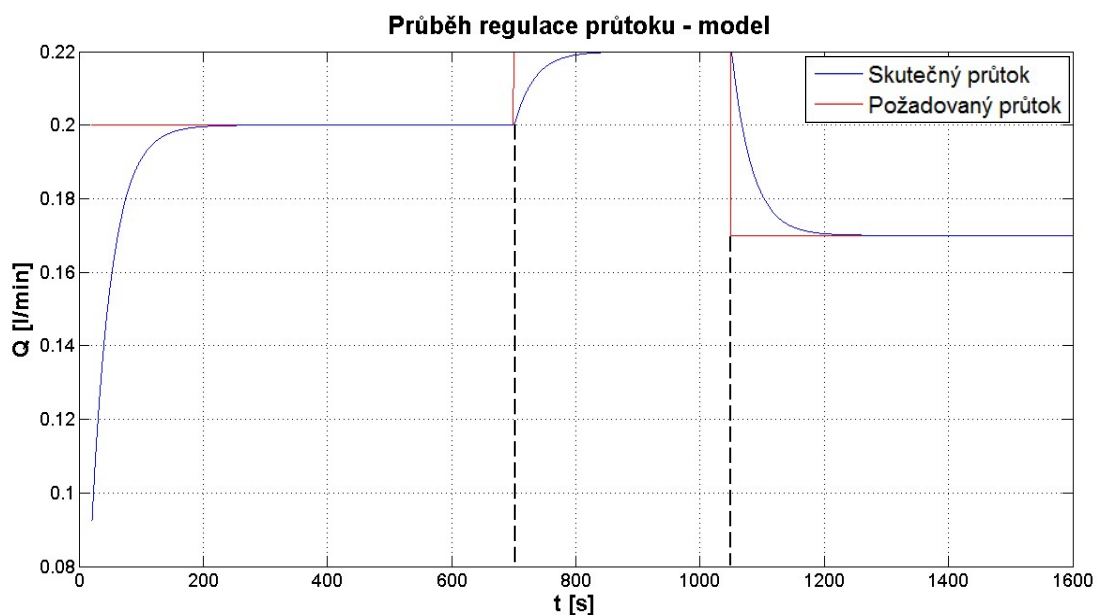
Řešení a následná realizace se pro tento případ omezí pouze na samotný model, zrealizovaný v programu Matlab & Simulink, viz obrázek 24. Dalším krokem je otestování funkčnosti modelu a reálného systému s využitím měřicí karty PCI 1711. Oproti předchozím kapitolám, se regulátor se dvěma stupni volnosti nebude implementovat do mikroprocesoru, jako tomu bylo v předešlé kapitole 3.5.2. Výsledný model celého systému je vidět na obrázku 24. Průtok je regulován klasickým zpětnovazebním PI regulátorem, jako tomu bylo i v předchozích případech. Z tohoto důvodu byly i zde zachovány nastavené parametry P_Q a I_Q . K regulaci teploty je využit již zmíněný PID regulátor s dvěma stupni volnosti s parametry $P_T = 8,60$, $I_T = 0,08$, $D_T = 20,50$ a s váhovými koeficienty b a c , které jsou rovny jedné. Parametry regulátoru byly do jisté míry získány pomocí takzvaného autotuningu v programu Matlab s následným ručním seřízením. Dalším krokem je otestování vhodného nastavení obou regulátorů a ověřit správnou funkci

2DOF regulátoru. Výsledkem z testování modelu je opětovné získání dvou grafů s průběhem regulace průtoku a teploty vody. Průběhy získané ze simulace jsou vidět z grafu 33 a 34.

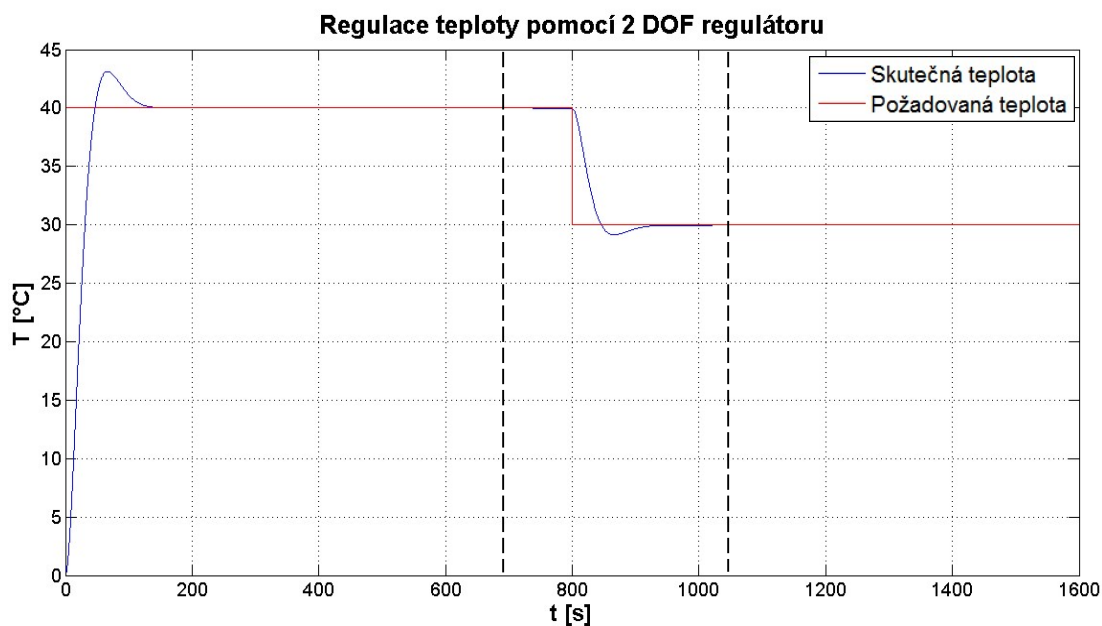


Obrázek 24: Model 2DOF regulátoru v programu Matlab & Simulink

Graf 33 představuje průběh regulace průtoku, který je i zde stejný a beze změny. Naopak druhý zmíněný graf, graf 34, který zobrazuje průběh regulace teploty pomocí 2DOF regulátoru, potlačil a eliminoval křížovou vazbu, která v tomto případě reprezentuje poruchu. Tudíž i toto provedení se ukázalo jako vhodné k eliminaci křížové vazby. Ovšem ještě je nutné propojit model s reálnou úlohou pomocí měřicí karty PCI 1711 a celé zapojení otestovat a ověřit výslednou funkčnost nejen na samotném modelu.



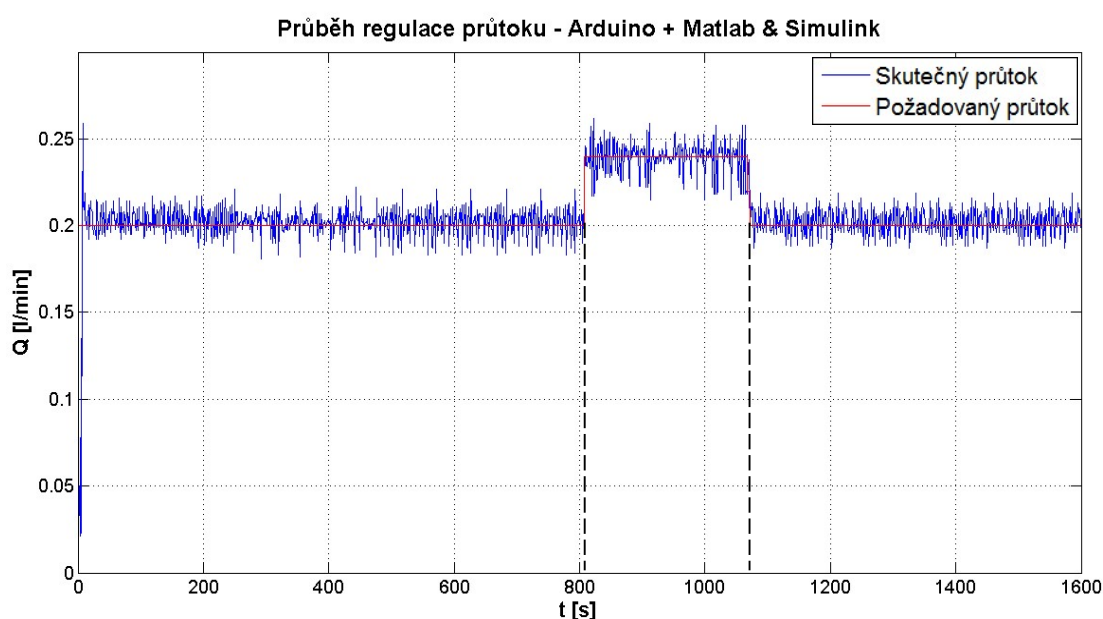
Graf 33: Průběh regulace průtoku – model



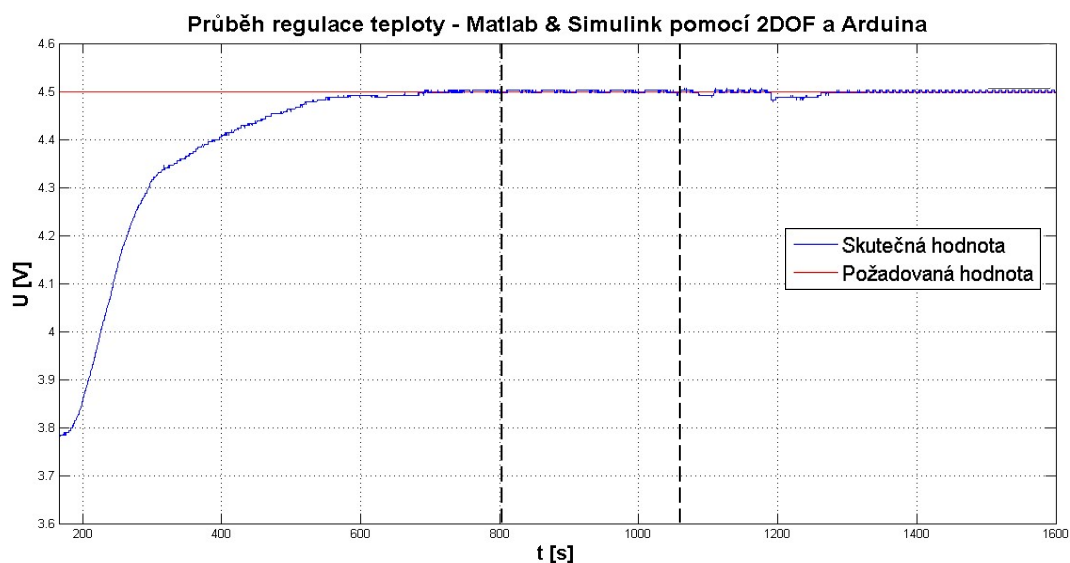
Graf 34: Průběh regulace teploty za pomoci 2DOF regulátoru

Stejně jako v experimentu z předchozí kapitoly, tak i zde se propojí model realizovaný v programu Matlab & Simulink se systémem pomocí měřící karty PCI 1711. Průtok se nejprve nastaví na počáteční žádanou hodnotu 0,20 l/min a teplota je nastavena na žádanou hodnotu 4,5 V. Po ustálení hodnot průtoku a teploty se provede první skoková změna průtoku na 0,24 l/min a následně se provede skoková změna průtoku zpět na

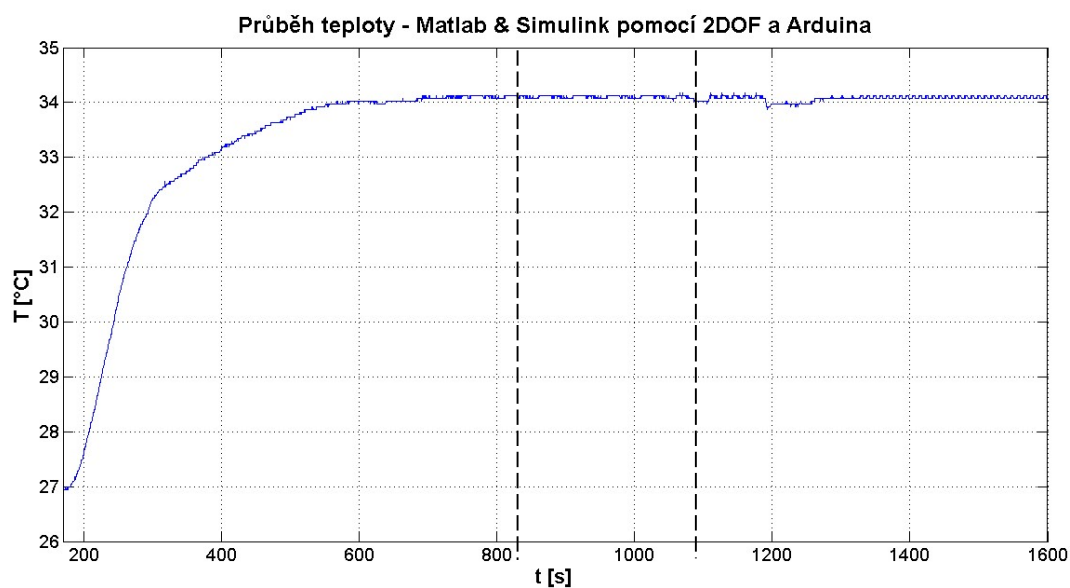
0,20 l/min. Výsledkem celého experimentu jsou dva grafy s průběhem regulace průtoku a teploty. Graf 35 zobrazuje průběh průtoku, který je regulován klasickým zpětnova-zebním PI regulátorem, jehož hodnoty P_Q a I_Q jsou nastaveny stejně, jako tomu bylo v předchozích kapitolách. Graf 35 nepřináší žádnou zásadní změnu, avšak následující graf, který zobrazuje regulaci teploty pomocí 2DOF regulátoru, již změnu přináší. Z grafu 36 je vidět, že regulace teploty pomocí regulátoru s dvěma stupni volnosti dokáže rovněž potlačit a eliminovat negativní vliv křížové vazby, čímž nedochází k ovlivnění teploty změnou velikostí průtoku. Současně s tímto měřením, byla teplota měřena nejen PT100, ale i teplotním čidlem - Arduino. Výsledný průběh je vyneseno do grafu 37, kde je teplota ve stupních Celsia. Průběhy teplot a napětí z grafu 36 a 37 jsou totožné a mají i stejný průběh.



Graf 35: Regulace průtoku pomocí PI regulátoru a Arduina



Graf 36: Průběh regulace teploty pomocí modelu s 2DOF modelovým regulátorem



Graf 37: Průběh regulace teploty naměřený teplotním čidlem s Arduinem

Při porovnání výsledků z předchozích kapitol je vidět, že i tato metoda vede k potlačení negativního vlivu křížové vazby, stejně jako tomu bylo v modifikaci s filtrem. Modifikace využívající 2DOF regulátor teploty je rovněž vhodná a funkční. Další možností je pozměnění parametrů regulátoru s dvěma stupni volnosti a tak docílit kratší doby, potřebnou k dosažení žádané hodnoty. Ovšem i přesto je tato metoda vhodná a použitelná jako další způsob k potlačení a eliminaci křížových vazeb.

4 Vzdálená komunikace

Kapitola se zabývá využitím vhodné nadstavby včetně pokusu o vzdálenou komunikaci a popřípadě i možnost nastavení regulátoru pomocí internetu. K tomuto účelu je zde využito již výše zmíněné nadstavby v podobě Ethernet shieldu, který je na obrázku 27.



Obrázek 25: Modul - Arduino Ethernet shield

Z obrázku je vidět, že se jedná o podobnou desku, jako je Arduino samotné. Spodní část desky je opatřena piny, pomocí nichž se Ethernet shield nasadí na Arduino. Je třeba dodržet polaritu a správně shield nasadit. Tak dojde k zamezení nesprávné funkčnosti shieldu a výstupních pinů. Jednou z mnoha výhod takového provedení je jednoduchost a snadná manipulace v případě poruchy, kdy je možné jednoduše shield odejme v případě potřeby z Arduina. Další nespornou výhodou je, že s Ethernet shieldem se nepřichází o vstupní a výstupní piny, ve kterých je Ethernet shield nasazen, což lze vidět z obrázku 27.

Jedná se tedy o rozšiřující modul, s jehož pomocí se Arduino připojí k internetu. K síti jej lze připojit standartním kabelem s koncovkou RJ-45. Provozní napětí je 5 V, což je stejné jako u platformy Arduino. Využívá se zde čipu Wiznet W5100 s interní 16K pamětí. Rychlost připojení je 10 / 100 Mb. Komunikace mezi Ethernet shieldem a Arduinem je realizována pomocí SPI portu [4]. K dispozici je i slot pro micro-SD kartu, která může být použita pro ukládání dat či souborů. Následně můžou být tato data či soubory posílány nebo čteny přes síť. Shield dále podporuje současně až čtyři připojení najednou.

4.1 Monitoring regulace

Nejprve je třeba do hlavního programu vložit potřebné knihovny pro Ethernet shield. Následně je nutná volba MAC adresy (Media Access Control), což je jedinečný identifikátor pro síťové zařízení. Adresa přidělena od výrobce je vždy celosvětově jedinečná. Zde ovšem není MAC adresa přidělena hardwarově, a proto je zadávána dle libosti. Je možné využívat i vymyšlenou adresu, kterou musí tvořit šest hexadecimálních čísel. Další důležitou adresou pro komunikaci je IP adresa (Internet Protocol). Zjištěná IP adresa pro danou síť, ve které se shield nachází je ve tvaru 147. 230. 4. Z, kde Z představuje poslední trojčíslí, které je v rozmezí od 0 do 255. Přidělením IP adresy Ethernet shieldu umožní následné připojení přes internet. V programu to vypadá pak následovně. První řádek představuje výše zmíněnou MAC adresu, která byla zvolena následovně, viz zdrojový kód 9.

```
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };  
IPAddress ip(147, 230, 4, Z);
```

Zdrojový kód 9: Programové řešení MAC a IP adres

Následující řádek je pro IP adresu, kde je třeba za Z dosadit vhodné číslo (0 - 255). Zároveň pomocí IP adresy, kterou je nutno napsat do příslušného internetového vyhledávače, se lze připojit k Ethener shieldu. Ještě je nutné do smyčky s označením *Setup* vložit několik řádků, které spustí komunikaci a vypisování přes internet. Samotný program pro monitoring je ve funkci, která je volána při spuštění příslušné regulace. Po zapojení Ethernet shieldu a následném spuštění webového prohlížeče s příslušnou adresou, se na stránce objeví text s informací, zda a jaká regulace právě probíhá. Zda probíhá regulace průtoku, či zda probíhá regulace průtoku a teploty vody. Následně se vypíše příslušný text s informacemi pro příslušnou regulaci. Ukázkou je zdrojový kód 10. První řádek slouží k informaci o průběhu příslušné regulace. Následně se vykreslí text, který informuje o nastaveném průtoku a aktuálním průtoku. Výsledkem je možnost sledovat online regulace průtoku, konkrétně pro tuto ukázkovou část zdrojového kódu. Zároveň je nastavena velikost, barva i font písma, jak je ze zdrojového kódu 10 vidět. V případě, že by nastavené parametry nevyhovovaly požadavkům uživatele, je možné hodnoty parametrů kdykoli změnit, či vhodně upravit.

```

client.println("<font color='red' size='5'> <h3 = right>Nastaveny
prtok: ");
client.println(des);
client.println(" L/min ");
client.println("<font color='black' size='5'> <h3 = right>Aktualni
prtok: ");
client.println(Q);
client.println(" L/min </h3>");

```

Zdrojový kód 10: Ukázka kódu vykreslujícího data na internet

Výše popsaným způsobem je zde řešen monitoring pomocí internetu. Dalším krokem by bylo umožnit uživateli ke sledování regulované a nastavené veličiny i možnost měnit parametry regulátoru spolu s žádanými hodnotami. Ovšem v tuto chvíli to není možné a změna parametrů se provádí pouze v laboratoři na samotném zařízení. Programové řešení aktuálně neumožňuje změnu žádaných hodnot a parametrů obou regulátorů přes internet. Důvodem je nestandardní chování regulátorů při snaze měnit parametry pomocí internetového připojení. Proto bylo od toho upuštěno a zůstalo se pouze u sledování regulovaných hodnot přes internet. Výsledný efekt je pak možno vidět při spuštění celého zařízení. Pro představu je ještě přiložen obrázek 28, kde je možné si prohlédnout výše popsaný monitoring pomocí internetu. V úvodu je nadpis celé stránky. Dále je text informující o typu zvolené regulace. Následují informace o velikosti příslušné proporcionální, integrační a případně i derivační složce. Jako poslední je zde informace o velikosti nastavených hodnot a informace o velikosti aktuálních hodnot průtoků a teplot.

Arduino WEB server

Monitoring

REGULACE PRUTOKU

P= 0.05 ; I= 1.80

Nastaveny prtok: 0.25 L/min

Aktualni prtok: 0.00 L/min

Obrázek 26: Webová stránka pro vzdálený monitoring

5 Kompletace zařízení

Před finálním zkompletováním celého zařízení předcházelo zkoušení jednotlivých prvků samostatně, tedy každého prvku zvlášť. Poté se začaly prvky postupně spojovat, až do finální podoby. Je to z toho důvodu, aby nedošlo k nežádoucímu poškození již funkčních částí. Zároveň se tím eliminují možné chyby a závady na zhotovených částech. Po zkompletování a odzkoušení zařízení je třeba finální celek uložit do ochranného boxu. Pro tyto účely byl vybrán malý černý box s hliníkovými čely od firmy TEKO, který je na obrázku 29.

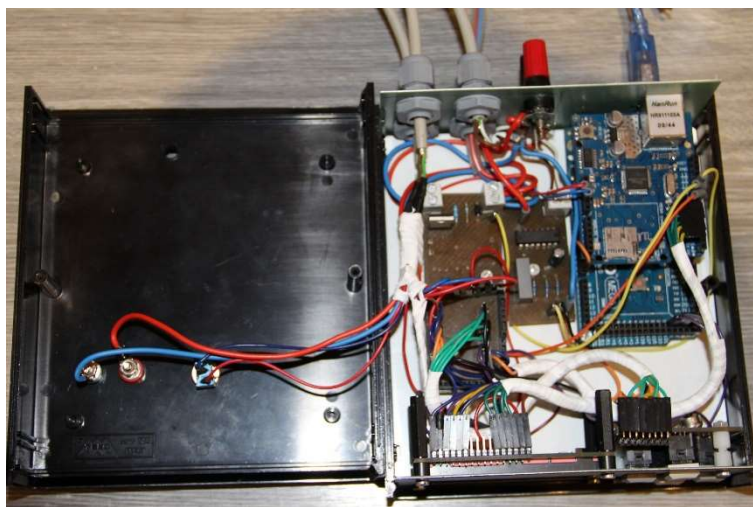


Obrázek 27: Plastový box

Spodní a vrchní část boxu je k sobě připevněna dvěma šrouby. Po vymontování šroubů je možné odklopit vrchní víko a podívat se do vnitřku boxu. Jednotlivé části zařízení jsou uloženy vně boxu s průchodkami pro vodiče a se vstupními a výstupními svorkami pro napěťové signály. V přední části boxu je LCD displej s ovládacími tlačítky. Horní část boxu je opatřena polohovacím přepínačem ON/OFF a výstupními svorkami pro napěťový signál, který je v rozmezí 0- 10 V dle velikosti průtoku. Ze zadní části boxu jsou vyvedeny čtyři vývodky se čtyřmi vodiči. Dále se zde nachází vstup, do kterého je připojen napájecí kabel pro Arduino. Hned nad ním je vstup pro Ethernet shield, kam se připojí standardní datový kabel s RJ-45 koncovkou. Zároveň se zde nachází i svorky pro napájecí napěťový signál o velikosti 12 V.

Dno boxu je opatřeno deskou z umělé hmoty, která tvoří takzvané falešné dno boxu. Na falešném dnu jsou uchyceny jednotlivé desky plošných spojů pomocí sloupků a samořezných šroubků. Takto upevněné desky je možné při poruše snadno vyjmout a provést případnou opravu. Zároveň se při přenášení boxu, či nežádoucích otřesech nebudou

desky uvnitř volně pohybovat. Propojení jednotlivých desek mezi sebou je realizováno pomocí barevných vodičů, což umožňuje přehlednost a případné změny při volbě a výběru pinů. A to ať už jde o Arduino piny či o piny, které jsou určené pro napájení. Celé zapojení je koncipováno do takzvaného stavebnicového provedení. Způsob provedení je vidět na obrázku 30. Zároveň to umožňuje snadný zásah do zařízení v podobě žádaných úprav nebo potřebných oprav, které by mohlo zařízení v průběhu času vyžadovat.

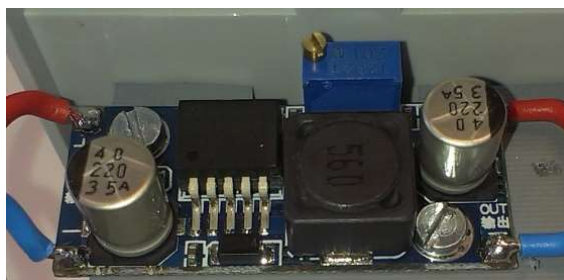


Obrázek 28: Pohled do boxu

V pravém rohu boxu, na obrázku 30, je uložena platforma Arduino s nasazeným Ethernet Shieldem. Vlevo od Arduina je deska obsahující výkonové obvody pro spínání čerpadla a topného tělesa. Zároveň jsou součástí desky dvě řady pinů určené pro napěťový signál o velikosti 5 V, který je vyveden z Arduino desky. V horní části obrázku 30 jsou vidět vývodky s vodiči, napájecí svorky a napájecí kabel k Arduino a přípojka pro kabel s RJ-45 konektorem. Ve spodní části obrázku 30 je vidět připojení LCD displeje. Vedle displeje se nachází deska plošných spojů osazená tlačítky, která jsou určena pro ovládání a pohyb v menu. V horním víku jsou pak napěťové svorky s výstupním napěťovým signálem a přepínačem s polohou ON/OFF pro zapnutí či vypnutí celého zařízení. Přepínač zde spíše funguje jako nouzový vypínač pro situaci, kdy bude nutné z nějakého důvodu zařízení vypnout.

Zařízení se napájí stejnosměrným napětím o velikosti 12 V. Jedním z požadavků je, aby bylo možné zařízení napájet z pevného zdroje napětí, který je pro tento typ úlohy určen.

Výstupní napětí, které je možné ze zdroje dostat je 5 V, 9 V, 12 V a 15 V. Proto se mezi zdroj napětí a zařízení vloží stabilizátor napětí zobrazený na obrázku 31. Jedná se o stabilizátor napětí s typovým označením 3A step-down DC/DC měnič napětí obsahující konvertor LM2596. Vstupní napětí, které se smí přivést na vstup stabilizátoru je v rozmezí 4 V až 35 V [10].



Obrázek 29: Stabilizátor napětí na 10 V

Výstupní napětí je pak v rozmezí od 1,25 V do 32 V s možností nastavení pomocí trimru. Doporučená hodnota vstupního proudu jsou 2 A a maximální dovolená hodnota proudu jsou 3 A. Nezbytná součást obvodu, která je v podobě chladiče, slouží pro odvod tepla ze stabilizátoru do okolí. Úbytek napětí na stabilizátoru jsou 3 V, proto se na vstup stabilizátoru přivádí napětí o velikosti 15 V. Destička se stabilizátorem a chladičem je uložena do krabičky se vstupními a výstupními vodiči. Vodiče jsou navíc barevně označeny červenou barvou pro kladný pól a modrou barvou pro záporný pól. Zároveň je krabička polepena a označena štítky s informacemi o vstupu, výstupu a polaritě vodičů, viz obrázek 32. Z přední i ze zadní části je krabička navíc ještě opatřena žebrováním, které umožní proudění vzduchu, a tak dochází k odvádění tepla z chladiče do okolí. Výstupní vodiče ze stabilizátoru jsou navíc ještě opatřeny banánky, které se dají připojit na vstupní svorky boxu, ve kterém je zařízení uloženo. Vstup do stabilizátoru se připojí na napěťový zdroj s výstupními svorkami, které jsou označeny příslušnou hodnotou napětí. V tomto případě jde tedy o napájecí napětí o velikosti 15 V.



Obrázek 30: Krabice pro stabilizační obvod s chladičem

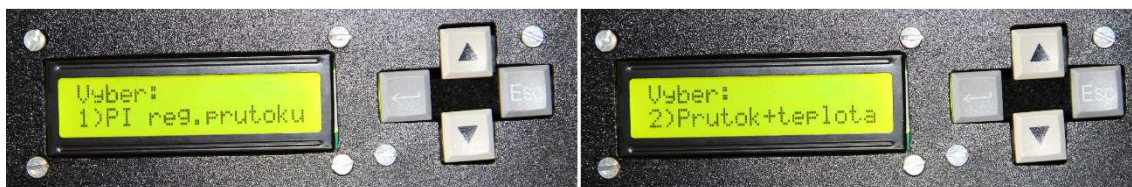
Všechny svorky a vodiče jsou označeny štítky s příslušným potiskem, což usnadňuje práci a manipulaci se zařízením. Dále štítky s informací napomáhají snižovat rizika poškození zařízení při manipulaci a připojování napěťového signálu na správné svorky. Tak se usnadní uživateli práce se zařízením. Současně je zde snaha o to, aby vše bylo srozumitelné a pochopitelné pro uživatele, který přijde se zařízením poprvé do styku. Ovšem stále je zde předpoklad, že se zařízením bude pracovat uživatel, který má alespoň nějaké znalosti pro práci se zařízením a jeho obsluhou. Dále pak jisté znalosti ohledně regulace (nastavování regulátorů).

6 Ovládání a obsluha zařízení

Tato kapitola může být brána i jako manuál pro práci se zařízením a jeho obsluhu. Před samotným spuštěním je třeba zkontrolovat, zda je vše zapojeno tak, jak má být. To znamená, že je připojeno napájecí napětí 12 V na přívodních svorkách, které jsou v zadní části boxu. Není-li k dispozici zdroj s napájecím napětím 12 V, je třeba využít stabilizátor, který je napájen stejnosměrným napětím 15 V a výstupní napětí ze stabilizátoru se přivede na vstupní svorky zařízení. Zkontrolovat zda je připojeno čerpadlo pomocí příslušné svorky. Dále zda je teplotní čidlo v nádobě s topným tělesem na správném místě a zda je připojen kabel s výstupním signálem 0 V až 10 V na napěťově řízený zdroj napětí, který mění toto vstupní stejnosměrné napětí na střídavé napětí v rozmezí 0 V až 230 V a napájí topné těleso. Zároveň je třeba zkontrolovat a popřípadě správně zapojit průtokoměr. Dále připojit síťový kabel, čímž se umožní monitorování regulace přes internet. Jako poslední krok zbývá připojit napájecí kabel k platformě Arduino a přepnout vypínač do polohy zapnuto (ON). Po zapnutí zařízení se rozsvítí LCD displej s příslušným textem. Čtyři tlačítka na přední straně boxu umožňují pohyb v menu s následnou volbou typu regulace. Po výběru typu regulace následuje volba příslušných parametrů. Kompletní řešení pohybu v menu včetně stromu je vyobrazeno v příloze C. Nejprve pomocí šipek nahoru a dolu se zvolí způsob regulace. Tlačítkem *Enter* se potvrdí volba výběru. Následuje výběr parametrů (např. průtoku nebo jednotlivých složek PID regulátoru). Opět pomocí šipek nahoru a dolu se mění velikost parametrů a následným stiskem tlačítka *Enter* se opět potvrdí výběr. Změna předchozího parametru se provádí kombinací tlačítek šipka nahoru a tlačítka *Esc* současně. Po nastavení a potvrzení všech proměnných se spustí samotný program. Pokud je při běhu programu stisknuto tlačítko *Enter*, dojde k návratu do části menu, která je určena pro výběr parametrů proměnných. Stiskem tlačítka *Esc* dojde k návratu do základního menu, kde je možné opětovný výběr či změna typu regulátoru.

Při prvním spuštění regulace se může stát, že vodní soustava nebude zaplněna vodou. Proto je dobré čerpadlem vehnat vodu do soustavy. Jednak nenastane situace, kdy v nádobě s topným tělesem nebude voda a pak se vyžene veškerý přebytečný vzduch z vodní soustavy. Tento přebytečný vzduch zamezuje plynulé regulaci průtoku a znehodnocuje měření. Dosažení celkového zavodnění soustavy je zde řešeno opět kombinací dvou tlačítek. A to stiskem tlačítka s šipkou nahoru a šipkou dolu. Jsou-li obě tyto tlačítka stisknuta současně, přivede se na čerpadlo konstantní hodnota napětí a to začne

hnát rychle vodu z nádoby. Tím dojde k vyhnání přebytečného vzduchu z vodní soustavy pryč. Ve chvíli, kdy se tlačítka uvolní, nebo nebudou stisknuta, bude čerpadlo opět fungovat tak, jak se od něj očekává pro daný regulační pochod.



Obrázek 31: Pohled na displej s úvodním menu

Pro úplnost je zde ještě přiložen ilustrační obrázek 33, kde je vidět zmiňovaná přední (čelní) strana boxu s displejem a tlačítky. Současně je vidět první a druhý obraz výběru základního menu, který je možné měnit a vybírat díky tlačítkům. Po stisku tlačítka *Enter* se spustí nastavování, které již bylo popsáno v předchozím odstavci.

7 Závěr

V rámci diplomové práce byla řešena problematika regulace průtoku a teploty vody. Nejprve byla řešena problematika spojená s regulací samotného průtoku. Hlavním bodem při řešení tohoto problému, byla realizace číslicového regulátoru, který byl řešen za pomoci platformy Arduino Mega 2560. Vycházelo se z vhodně zvolené podoby regulátoru, která byla následně naprogramována a implementována do platformy Arduino. Následovalo odladění programu a testování celku v laboratorních podmínkách, ale i ve školní laboratoři. Funkčními testy se prokázala funkčnost zařízení. Program reagoval na změny vstupních parametrů. Vstupní parametry byly programu zadány již při spuštění. Dále se tyto parametry měnily pomocí tlačítek. I na tyto změny program reagoval dle očekávání s kladnou odezvou. Jako problematičtější se ukázala přesnost průtoku. Jak již bylo řečeno, při konstantním napětí na čerpadle se nedostává konstantní délka pulzů. Právě rozdílné délky měly za následek lišící se periody. To mělo za následek rozdílné frekvence a tím pádem i průtoky. Odchyłky nebyly nijak velké, ovšem při malých průtocích byly bohužel patrné. Rozdíl pak byl o velikosti $\pm 0,015$ l/min. Pro větší přesnost by bylo možné nahradit stávající průtokoměr nějakým přesnějším průmyslově vyráběným průtokoměrem, čímž by se zvedla přesnost měřeného průtoku.

Obdobně byl realizován i další bod zadání, kdy bylo řešeno provedení a realizace jed nouúčelového regulátoru průtoku a teploty kapaliny. Opět se vycházelo z programově řešeného regulátoru z předchozí kapitoly. Jen s tím rozdílem, že zde byly tyto regulátory dva. To umožnilo regulovat průtok, ale i teplotu vody najednou. Současně s tím byl řešen i problém s takzvanými křížovými vazbami. Regulátor, který řídí vlastní prvek soustavy, svým akčním zásahem ovlivní i stav druhého prvku. Experimentálními měřeními, které proběhly rovněž v laboratoři, byla prokázána přítomnost křížové vazby. Zároveň bylo odzkoušeno celé zařízení v praxi, tudíž se odzkoušela regulace průtoku a teploty současně. Čímž byla prokázána funkčnost celku i při tomto typu regulace. Nakonec byly realizovány některé metody, které vedly k potlačení a eliminaci křížové vazby a zlepšení regulačního pochodu.

Součástí bylo i programové řešení takzvaného menu, v němž se dá pohybovat a vybírat požadované parametry. První možností byla regulace průtoku a druhá byla regulace průtoku a teploty vody. Následná možnost výběru parametrů a spuštění běhu programu. Vše se zobrazuje na LCD displeji. Dále zde byl řešen monitoring příslušné regulace

přes internet pomocí Ethernet shieldu, což umožnilo sledovat příslušný typ regulace z jiného počítače. Byla vidět jak data zadaných hodnot, tak i data hodnot, které se průběžně měnila. Sledovaný výstup byl ovšem časově zpožděný. To bylo dáno obnovovacím časem internetové stránky, na které byla data zobrazována. Ovšem jednalo se o časový úsek, který nebyl delší než jedna sekunda. Tím se prokázala funkčnost monitoringu prostřednictvím internetu.

Dalším krokem, který by mohl být, je návrh a implementace 2DOF regulátoru do Arduina a umožnit tak výběr metody pro eliminaci křížové vazby. Zda bude uživatel chtít využít modifikace využívající filtr, nebo modifikace využívající regulátor s dvěma stupni volnosti. Současně by se mohl přidat ještě třetí regulátor, který by zajišťoval doplňování vody do nádoby s čerpadlem. Tím by odpadla potřeba hlídat stav vody v nádobě a nutnost ručního doplňování. Ovšem ještě by bylo nutno obstarat vhodný vodní ventil, který by byl ovládán již umíněným PID regulátorem, čímž by celá úloha stala plně automatizovaná. Programová část by se rozrostla o další regulátor a zároveň by se mohla rozšířit i o schopnost ukládat naměřená data na paměťovou kartu pro další zpracování či rovnou data vykreslovat do grafu v reálném čase. To vše by pak bylo možno sledovat, ale i ovládat prostřednictvím internetu, bez potřeby být přímo ve školní laboratoři, což by mohlo být náplní dalšího řešení a další možné práce.

Současně jsem se seznámil s prací a programováním platformy Arduino Mega 2560. Práce s programováním v jazyce, který je obdobou programovacího jazyka C++, který byl do té doby pro mě nepříliš známý a používaný. Stejně jako práce s Ethernet shieldem, kde se využívá programovací jazyk v podobě HTML (HyperText Markup Language). Zároveň jsem blíže porozuměl a osvojil si práci s proporcionálně-integračně-derivačním regulátorem.

Seznam doporučené a citované literatury

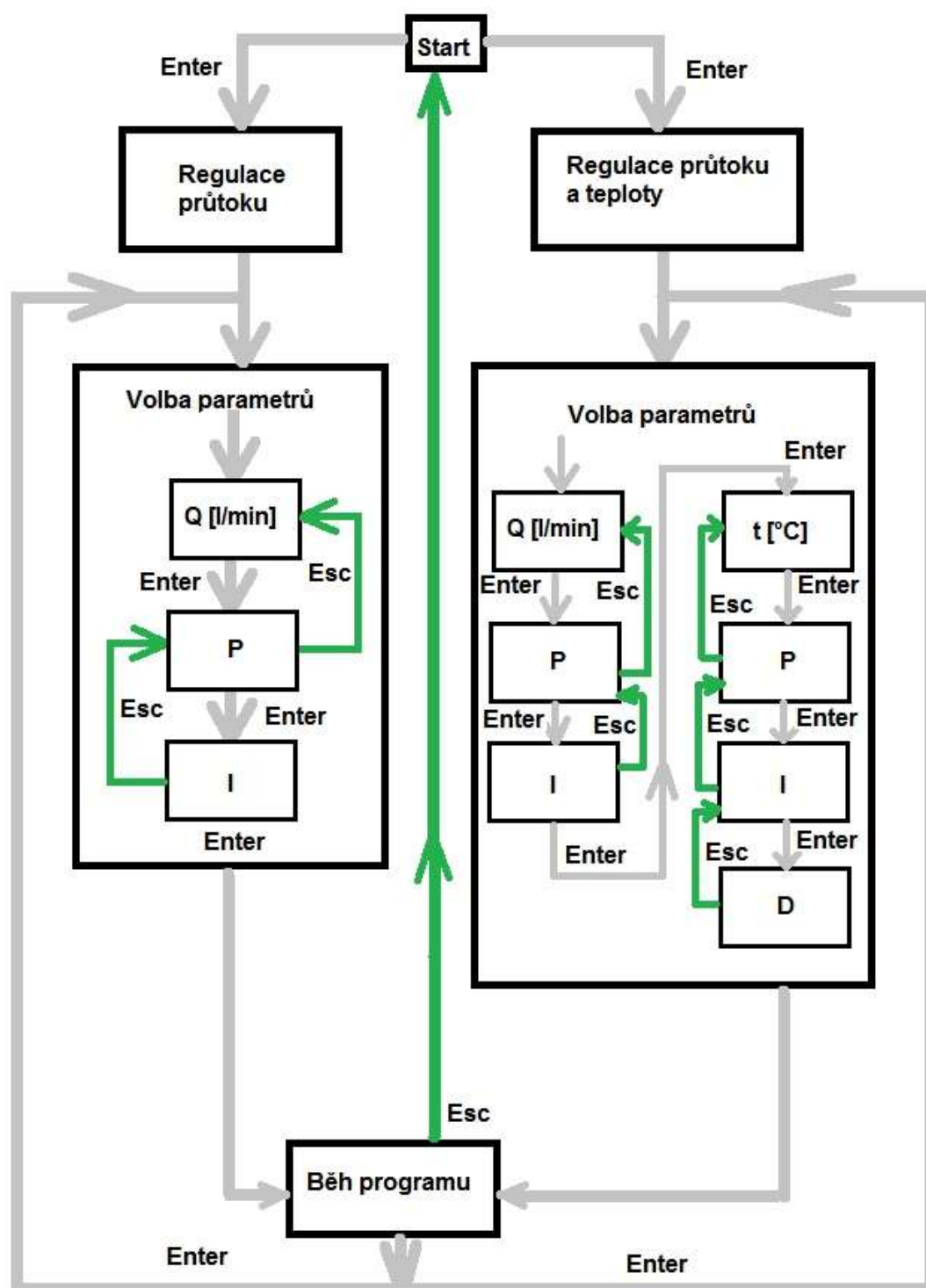
- [1] Arduino.cz – Webový magazín o Arduinu a elektronice [online]. 2014 [cit. 2014-10-07]. Dostupné z: <http://arduino.cz/>.
- [2] Arduino –Home [online]. 2014 [cit. 2014-10-07]. Dostupné z: <http://arduino.cc/>.
- [3] Arduino-products. Arduino board Mega 2560 [online]. 2015. vyd. [cit. 2015-05-04]. Dostupné z:<http://www.arduino.cc/en/Main/ArduinoBoardMega2560>.
- [4] Arduino-products. ArduinoEthernet shield [online]. 2015. vyd. [cit. 2015-05-04]. Dostupné z:<http://www.arduino.cc/en/Main/ArduinoEthernetShield>.
- [5] Electronic Kits. Thermistor [online]. 2014 [cit. 2015-04-25]. Dostupné z: http://www.eidusa.com/Electronics_Kits_TEMP_THERMISTOR_1.htm.
- [6] ČERNÍK, Martin. Řízené součástky [online]. Liberec, 2013 [cit. 2015-05-04]. Učební text. Technická univerzita v Liberci.
- [7] ELWEB.CZ. MOSFET tranzistory [online]. 2015 [cit. 2015-05-04]. Dostupné z: <http://www.elweb.cz/clanky.php?clanek=94>.
- [8] HLAVA, Jaroslav. Prostředky automatického řízení II. Praha: ČVUT, 2000.
- [9] MODRLÁK, Osvald a Lukáš HUBKA. Automatické řízení: učební text. Vyd. 1. Liberec: Technická univerzita v Liberci, 2012.
- [10] LM2596 DC-DC Step-Down Module BK_DC010. 2011. ELECFREAKS [online]. [cit. 2015-05-06]. Dostupné z: <http://www.electfreaks.com/store/lm2596-dcdc-stepdown-module-bkdc010-p-292.html>.
- [11] Ponorné čerpadlo. Barwig Typ 04 [online]. 2015. vyd. [cit. 2015-05-04]. Dostupné z: <http://www.conrad.cz/ponorne-cerpadlo-barwig-typ-04-0444-12-v-dc-0-9-1-7-a-10-l-min-6-m.k539090>.
- [12] Průtokoměr FCH-m-POM-LC. 2010. Dostupné z: <http://www.produktinfo.conrad.com/datenblaetter/150000-174999/150391-an-01-cs-Prutokomer.pdf>.

- [13] Steinhart–Hart equation. 2001-. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation [cit. 2015-05-06]. Dostupné z: http://en.wikipedia.org/wiki/Steinhart–Hart_equation.
- [14] ÅSTRÖM, Karl J, Tore HÄGGLUND a Karl J ÅSTRÖM. PID controllers. 2nd ed. Research Triangle Park, N.C.: International Society for Measurement and Control, c1995, viii, 343 p. ISBN 1556175167.

A Obsah přiloženého CD

- text diplomové práce
 - diplomova_prace_2016_Jan_Sec.pdf
 - diplomova_prace_2016_Jan_Sec.doc
- fotografie a ilustrační obrázky
- zdrojové kódy
 - hlavní program (v programovém jazyce Arduino)
 - PID regulátor (v programovém jazyce Arduino)
 - filtr (v programovém jazyce Arduino)
- katalogové listy použitých součástek
- naměřená data (v tabulkovém editoru Microsoft Excel)

C Provedení stromu menu



D

Schéma zapojení celého zařízení

